

Grado Ingeniería Informática
2016-2017

Trabajo Fin de Grado

Generación automática de Narrativas

Alberto Suárez Rico

Tutor

Daniel Borrajo Millán



Esta obra se encuentra sujeta a la licencia Creative Commons **Reconocimiento – No Comercial – Sin Obra Derivada**

Summary

The text reproduced below is a summary consisting of an introduction of the work that has been done, the main objectives, the results and the conclusions. The following summary does not try to be a complete description of the system but a short introduction to it.

Introduction

Artificial intelligence

In 1956 John McCarthy defined artificial intelligence as “the science and engineering of making intelligent machines”. Nowadays the artificial intelligence or computing intelligence is one of the main branches studied by computer science, and it is defined as “the intelligence shown by machines”, or in other words, it is the capacity that machines have to learn and solve problems.

Currently the artificial intelligence has a wide variety of application fields, as for example:

- Artificial life
- Automatic thinking
- Biological computing
- Knowledge representation
- Semantic web
- Anti-spam mail
- Robotics
- Hybrid systems
- Intelligent agents
- Intelligent control

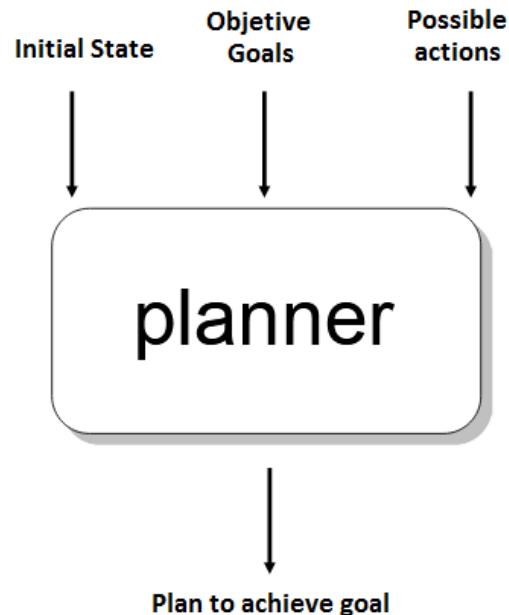
One of the main objectives of artificial intelligence is giving computers the capacity of imitating the logical thinking of human beings and they can be used as well to solve problems following the same reasoning that a normal person would use to it. The advantages of using computers to solve people’s every-day problems are:

- Significant time reduction in the solution of a problem.
- Elimination of human error.

One of the problems in which the use of computers can be very useful is the task planning. Planning is a thinking and reflection process consisting of choosing and organizing actions anticipating their results. The objective of this reflection is to achieve, as efficiently as possible, one or more pre-established targets. With the use of computers it is possible to automate this task of planning.

Automatic Planning

On this Final Degree Project we have used the automatic planning to generate narrative stories. The automatic task planning is a discipline of the artificial intelligence that consists of generating automatically an action plan when there is a problem to solve and, after following this plan, you will reach the solution of the given problem. The action plan is generated thanks to the use of a planner.



We have to provide the planners with a representation of the final state that we want to achieve, a representation of the detailed actions that can be taken and a representation of the beginning point. Moreover, we have to provide these planners with the capacity of generating a plan to achieve the target. A plan is a sequence or an action list that goes from a beginning point to a final state. In order to provide the planner with all these elements we have to implement:

A domain with the definition of:

- Objects and predicates to represent the states
- Actions to modify the states

A problem with the definition of:

- The beginning point
- The final state to be achieved

Planning language PDDL1.2

On this Final Degree Project, both the domain and the problems have been implemented with the planning language PDDL1.2 (PDDL for Planning Domain Definition Language).

PDDL is an attempt to standardize the planning languages. PDDL was developed by Drew McDermott and his colleagues in 1998. This language was inspired by other planning languages such as STRIPS and ADL.

On this work we have used PDDL language 1.2 version.

In PDDL 1.2 the planning model is divided into domain and problems. In the below lines we are going to describe the structure of every part of the planning model.

Domain

The structure provided by a PDDL 1.2 domain is as follows:

```
define (domain DOMAIN_NAME)(  
    (:requirements [:strips] [:equality] [:typing] [:adl])  
    (:types obj1 obj2 – obj  
        ...  
    )  
    (:predicates ( PREDICATE_1_NAME [?A1 ?A2 ... ?AN])  
        (PREDICATE_2_NAME [?A1 ?A2 ... ?AN])  
        ...  
    )  
    (:action ACTION_1_NAME  
        [:parameters (?P1 ?P2 ... ?PN)]  
        [:precondition PRECOND_FORMULA]  
        [:effect EFFECT_FORMULA]  
    )  
    (:action ACTION_2_NAME  
        ...  
    )  
    ...  
)
```

- Define: The first thing that we have to do is to give name to the domain.
- Requirements: After that, we explain the PDDL requirements that we can be used on the domain.
- Types: Here we explain the hierarchy of the objects that will take part in the actions.
- Predicates: In this part of the domain we define the predicates that will be used to describe the states.
- Actions: Eventually we set out in detail the different actions that the planner will be able to execute in order to modify the states.

Problem

The structure of a PDDL 1.2 problem is explained below:

```
(define (problem PROBLEM_NAME)
  (:domain DOMAIN_NAME)
  (:objects OBJ1 OBJ2 ... OBJ_N)
  (:init ATOM1 ATOM2 ... ATOM_N)
  (:goal CONDITION_FORMULA) )
```

- Define: The first thing is to give name to the problem.
- Domain: After that we mention the domain in which the problem has to be executed.
- Objects: Following that, in this part of the problem we have to decide the possible values that the objects defined on the domain can have.
- Init: In this section we have to define the starting state. In order to define this starting state we use the predicates that are defined on the domain.
- Goal: Finally, we set out the objectives we want to achieve.

Sayphi Planner

In order to execute the planning model that we have created on this work we have used the Sayphi planner.

Sayphi is an automatic planner based on searching-forward patterns such as FF. Furthermore, Sayphi does include automatic learning techniques that allow us to improve the results.

Among the qualities of the Sayphi planner we can say that the main areas to be highlighted are the two properties that make Sayphi a suitable planner for our problem:

- Sayphi is independent from the domain because its code does not make any reference to any domain. By using Sayphi planner we can execute any domain and generate plans.
- Sayphi, thanks to its searching techniques, can generate plans in a pseudorandom way, choosing actions determined by implemented planning mechanisms or choosing actions with a certain randomness. This randomness can be made using rrt algorithm.

Narration Fields

The narrative stories that this project generates have to do with the civil war simulated on the video game Far Cry 4. This topic is an ideal place to implement the automatic task planning, because, as it is usual on this kind of video-games, the characters are divided into two well-defined sides and the video game dynamics is to control the one-team characters and eliminate the opponent characters by carrying out several missions.

The aim of our project is, given the topics of Far Cry 4 video game, to plan the assignment of missions to the friendly-side characters and manage, by planning them, the different actions that will allow us to complete the missions, and following this path, generate a range of narrative stories with a variety of different ways to solve the video game.

Project Structure

In order to finish our work we have taken the following steps:

Firstly, we made a draft of the problem that we were dealing with, searching information about:

- Automatic planning
- PDDL programming language
- The “sayphi” planner
- The essays related to this topic

Secondly, we worked on the objectives that we were trying to achieve with our work.

Next, we made a specification of the functional and non-functional requirements that the system was going to support.

Following that, we began the system design by defining, on the one hand the planning domain with the objects, predicates and actions that would take part into the stories and, on the other hand, the problems that we would face.

After that, the experimental stage began, in order to analyze the working of the designed planning system.

And eventually we made the documentation of the project.

Objectives

The main objective of this project is to manage to generate automatically coherent and interesting narrative stories by using automatic task planning. These narrative stories will be obtained as a result of executing a domain on a planner. The domain will have to specify the different types of

- objects
- predicates
- actions

As we have already said, the topic of our domain will be about the civil war on the Kirat region that is simulated on the video game Far Cry 4. We have tried to make stories with a big amount of characters playing on them; they will be performing different missions moving from one place to the others, fighting against their enemies, obtaining and destroying buildings, being rewarding with powerful guns, etc.

One more objective that we have tried to achieve on this Final Degree Project is to analyze how the stories generated by the implemented domain are affected by the fact that the planner has the capacity of selecting domain actions randomly with a certain probability.

Summing up, the main objectives that we try to achieve on this project are the following:

- To design a domain of planning narrative stories with the theme of the Far Cry 4 video game.
- To arrange a wide variety of places, characters, guns, buildings and missions to the stories.
- To give coherence and interest to these stories, in spite of the fact that the planner is given the possibility to choose actions randomly.
- To create new stories with the support of the stories firstly generated.

The main motivation of this work is to manage that the stories generated by the implemented domain could be used by different companies related to entertainment, such as:

- Television channels, in order to produce their own programs, series or cartoons.
- Film production companies, providing them with an automatic method to elaborate the scripts for their films.
- Video-game companies: they could use this technique to create the video-game stories and, due to the big amount of stories that can be generated about one topic, they could create a unique experience for every user, and they could provide the games with different game levels.
- Publishing houses: they could use the implemented domain to create simple book stories or comics.

System design

In order to design the system we had to implement the domain with the PDDL 1.2 planning language.

And moreover, to try out how the domain works we had to create various problems, also with the PDDL 1.2 language.

And finally we have created the sentences that will be shown in the generated stories.

Domain

In order to implement the domain we had to design

- A structure of objects.
- Predicates
- Actions

Definition of the structure of objects

This definition has two objectives:

- Objects need to be useful and they can be used correctly to model the predicates of the domain. That is to say that the objects have to provide information to the different states of a problem and they need to have the capacity to be used in the actions defined on the domain.
- The structure of the objects must have a serious hierarchy in order to allow the actions to use the predicates consisting of objects with a low amount of instances and be able in that way to avoid unnecessary instantiations.

Definition of predicates

As predicates of the domain we have designed different types of information:

- Counting predicates: for representing numbered attributes for the guns or the characters than can be found in the stories.
- Place predicates: for placing the different characters on a map.
- Building predicates: for defining the states or attributes of the buildings in the story.
- Predicates for the characters: In the same way as we did for the buildings, we have defined predicates for the states or attributes of the characters in the story.
- Predicates for guns: to relate the guns to the characters.
- Predicates for missions: To make possible the relationship among different missions and to provide them with a description.

Definition of actions

The definition of actions is the main part of the domain. For obtaining interesting stories it is absolutely essential a varied definition of actions, with relevant preconditions and affecting a big amount of scene predicates. The actions of the domain can be divided into several parts.

- Actions for the allocation of missions: these are the actions that will allow the characters to assume or to order the missions.
- Actions for the allocation of characters' different ideologies: some missions will only be executed by characters from one of the factions and following one of the two ideologies of the game. With these actions the characters will become followers of the ideology they are not following.

- Actions that allow the characters to get guns. These actions will allow the characters to be willing to get guns, to get the guns that they do not have, to give guns to others and to destroy guns.
- Actions that allow the characters to attack the different buildings. Thanks to these actions the characters will be able to attack buildings, both for destroying or releasing them.
- Actions that allow the characters to do movements: this kind of actions are the main actions in the domain. These are the actions that allow the characters to do movements. With these actions the characters will be able to move from place to place. The characters will be able to move for several reasons such as for example:
 - To destroy or release a building.
 - To get a gun placed in a building
 - To give a gun to a different character of the story.
 - To get a gun from any other character.
 - To attack others.
- Actions that allow the characters to fight to each other with the guns: this kind of missions are essential on the implemented domain too. With these actions the characters can do different kind of attacks to each other.

Problems

In order to try out the domain we have designed 10 problems. Every problem has the same starting state, being the objectives the only difference between the problems.

Sentence creation

The actions implemented in the domain can be charged with a narrative meaning. In order to get that we have used the *narrate* function of the Sayphi planner; with the help of this function we can turn an action into a sentence. We use the action name and parameters to do that.

This is the way how, when creating a plan, we can manage to generate a sequence of sentences, easy to read and to understand.

Obtained Results

After finishing the experimental stage of the work, we can say that the achieved results have satisfied all the aims set out at the beginning.

We have managed to implement a domain that allows a task planner to automatically generate narrative stories. One thing has to be said though, these generated stories consist of a list of the events that happen in a story and they do not have a fluent narrative, in other words, the generated stories are a sequence of the actions taken by the different characters involved in the story: different kinds of actions can be provided, such as:

- Characters' movements
- Characters attacking buildings
- Attacks among characters
- Assignments of missions among characters

What follows is one of the stories generated by the Sayphi planner with the domain implemented on this work for solving a simple problem:

*SABAL consulta a AJAY si sería capaz de MATARAPAGAN, uno de los líderes que tiene oprimido al pueblo de Kirat, AJAY sin dudarlo se embarca en la misión
AJAY sabe que si quiere MATARAPAGAN necesitara tener en su poder un fuerte ARMAMENTO, por lo que quiere conseguirlo antes de ir a por PAGAN
El miserable TENIENTEPAGAN quiere conseguir el ARMAMENTO que podría acabar con su líder PAGAN
AJAY se dirige de BANAPUR a la FABRICA de SHANATH dado que allí se sitúa el eficaz ARMAMENTO que necesita
En la FABRICA de SHANATH, AJAY utiliza C4 para acabar con todos los soldados
En SHANATH, AJAY consigue que la FABRICA quede DESTRUIDA completando así la misión de DESTRUIRLAFABRICA
En la FABRICA de SHANATH. AJAY coge un mortífero ARMAMENTO
AJAY se dirige de SHANATH a FORTALEZA_PAGAN con el letal ARMAMENTO para acabar con PAGAN
En FORTALEZA_PAGAN, AJAY mata PAGAN utilizando el gran ARMAMENTO
End.*

A different objective that we intended to reach on this Final Degree Project is to analyze how the stories generated by the implemented domain can be affected by the fact that the planner where the domain is executed has the capacity of selecting domain actions randomly with a certain probability. After the experimental stage we have obtained the following results:

When the planner has a randomness probability of 0.001%, the planner generates a story that achieves all the proposed targets 100% of times. In this case the planner never chooses an action randomly.

When the planner has a randomness probability of 20%, the planner generates a story that achieves all the proposed targets 82'5% of the times. In this case the planner has a low probability of choosing an action randomly and most of the actions are chosen using a planning algorithm. That is why, even when the planner is sometimes choosing a random action and this fact takes it to the final state, it is highly probable the following actions to be chosen using a planning algorithm and so the final stage to be achieved.

When the planner has a randomness probability of 50%, the planner generates a story that achieves all the proposed targets 35% of the times. In this case the planner has a high probability of choosing an action randomly, and the planner –using its planning algorithm- might not choose any of the chosen actions.

When the planner has a randomness probability of 70%, the planner generates a story that achieves all the proposed targets 25% of the times. In this case the planner has a very high probability of choosing an action randomly and the planning task is very difficult to be done.

After obtaining these results we can conclude that the implemented domain is fitted with a wide range of actions and with these actions it can face different unexpected changes on the state.

Conclusions

Final comments

The main objective on this project has been to develop a system that allows us to create coherent narrative stories based on the automatic planning. After analyzing our results when doing the experimentation process we can now tell that this main objective has been achieved.

From a narrative point of view, the stories developed by the system are both coherent and interesting and, even if the system would be able to generate only a sequence of the events that take place in one of the stories, this sequence of events could be used to create a book story, a film script or a videogame plot.

The most difficult part of the work has been to achieve a very good command of PDDL language, because in the beginning my level of PDDL was not as high as needed and it took me quite a long time to be used to it. I found as well some difficulties on the process of modeling the planning domain, and I sometimes had to change, along the process, my decisions about the modeling.

Future lines

There are a lot of possibilities to improve this end-of-degree project.

In order to improve the quality of the narrative stories it could be possible to extend the domain code implementing it to PDDL3.0, and it could allow us to make some improvements on the narrative line by using “fluents” on the domain to model different attributes of the elements in the story, such as the characters’ life, the power of the guns and the number of soldiers that are defending the Kirat buildings.

We could also improve the translations of the domain actions to be able to produce stories easier to read and less repetitive.

In addition to that, some efforts could be done in order to improve the users’ experience, as for example:

- Creating an interface that allows the users to produce their own stories developing its problems by their own.
- Implementing a software to process the natural language reading the stories in a loud voice.

ÍNDICE DE CONTENIDO

1. Introducción	16
2. Estado del arte	18
2.1. Planificación automática de tareas.....	18
2.2. PDDL.....	18
2.3. Sayphi	23
2.4. Trabajos similares.....	24
3. Objetivo y motivación	26
4. Análisis del problema	27
4.1. Requisitos Funcionales	27
4.2. Requisitos No Funcionales.....	30
5. Diseño del sistema	31
5.1. Tema.....	31
5.2. Planteamiento	32
5.3. Definición de tipos de objetos	33
5.4. Definición de predicados	37
5.4.1. Predicado de Contadores	37
5.4.2. Predicados de Lugares	38
5.4.3. Predicados para Construcciones.....	39
5.4.4. Predicados para Personajes.....	40
5.4.5. Predicados para Armas.....	43
5.4.6. Predicados para Misiones.....	43
5.5. Definición de acciones.....	47
5.5.1. Misiones	48
5.5.2. Corrientes	53
5.5.3. Conseguir Armas	54
5.5.4. Ataques a construcciones	61
5.5.5. Movimientos	66
5.5.6. Ataques entre personajes.....	76
5.6. Creación de problemas.....	82
5.7. Creación de frases	87
5.7.1. Misiones	87
5.7.2. Corrientes	87
5.7.3. Movimientos	87
5.7.4. Conseguir armas	88

5.7.5. Ataques a construcciones	88
5.7.6. Ataques entre personajes.....	89
6. Experimentación	90
6.1. Plan de pruebas.....	90
6.2. Problemas	91
6.2.1. Problema 1	93
6.2.1. Problema 2	96
6.2.3. Problema 3	100
6.2.4. Problema 4	103
6.2.5. Problema 5	107
6.2.6. Problema 6	110
6.2.7. Problema 7	114
6.2.8. Problema 8	118
6.2.9. Problema 9	122
6.2.10. Problema 10	126
6.3. Análisis de los Resultados.....	131
7. Gestión del proyecto	133
7.1. Planificación	133
7.2. Planificación final	134
7.3. Presupuesto	135
7.3.1. Salarios	135
7.3.2. Personal.....	136
7.3.3. Materiales	136
7.3.4. Gastos Indirectos.....	137
7.3.5. Resumen.....	137
7.4. Entorno Socio-Económico.	137
7.5. Marco Regulador.	138
8. Conclusiones	139
8.1 Comentarios finales.....	139
8.2. Líneas futuras.	139
9. Referencias.....	140
10. Anexo: Código	141
Dominio.....	141

ÍNDICE DE TABLAS

Tabla 1: RF-001.....	27
Tabla 2: RF-002.....	27
Tabla 3: RF-003.....	27
Tabla 4: RF-004.....	27
Tabla 5: RF-005.....	28
Tabla 6: RF-006.....	28
Tabla 7: RF-007.....	28
Tabla 8: RF-008.....	28
Tabla 9: RF-009.....	28
Tabla 10: RF-010.....	29
Tabla 11: RF-011.....	29
Tabla 12: RF-012.....	29
Tabla 13: RF-013.....	29
Tabla 14: RF-014.....	29
Tabla 15: RNF-001	30
Tabla 16: RNF-002	30
Tabla 17: RNF-003	30
Tabla 18: PRE-001.....	38
Tabla 19: PRE-002.....	38
Tabla 20: PRE-003.....	38
Tabla 21: PRE-004.....	39
Tabla 22: PRE-005.....	39
Tabla 23: PRE-006.....	40
Tabla 24: PRE-007.....	41
Tabla 25: PRE-008.....	41
Tabla 26: PRE-009.....	41
Tabla 27: PRE-010.....	41
Tabla 28: PRE-011.....	42
Tabla 29: PRE-012.....	42
Tabla 30: PRE-013.....	42
Tabla 31: PRE-014.....	42
Tabla 32: PRE-015.....	43
Tabla 33: PRE-016.....	43
Tabla 34: PRE-017.....	44
Tabla 35: PRE-018.....	44
Tabla 36: PRE-019.....	45
Tabla 37: PRE-020.....	45
Tabla 38: PRE-021.....	45
Tabla 39: PRE-022.....	46
Tabla 40: PRE-023.....	46
Tabla 41: PRE-024.....	46
Tabla 42: Tabla Para Acciones	47
Tabla 43: ACC-001	48
Tabla 44: ACC-002	49
Tabla 45: ACC-003	50
Tabla 46: ACC-004	51
Tabla 47: ACC-005	52
Tabla 48: ACC-006	53
Tabla 49: ACC-007	53
Tabla 50: ACC-008	54

<i>Tabla 51: ACC-009</i>	55
<i>Tabla 52: ACC-010</i>	56
<i>Tabla 53: ACC-011</i>	57
<i>Tabla 54: ACC-012</i>	57
<i>Tabla 55: ACC-013</i>	58
<i>Tabla 56: ACC-014</i>	58
<i>Tabla 57: ACC-015</i>	59
<i>Tabla 58: ACC-016</i>	59
<i>Tabla 59: ACC-017</i>	60
<i>Tabla 60: ACC-018</i>	61
<i>Tabla 61: ACC-019</i>	62
<i>Tabla 62: ACC-020</i>	63
<i>Tabla 63: ACC-021</i>	64
<i>Tabla 64: ACC-022</i>	65
<i>Tabla 65: ACC-023</i>	66
<i>Tabla 66: ACC-024</i>	67
<i>Tabla 67: ACC-025</i>	68
<i>Tabla 68: ACC-026</i>	69
<i>Tabla 69: ACC-027</i>	70
<i>Tabla 70: ACC-028</i>	71
<i>Tabla 71: ACC-029</i>	72
<i>Tabla 72: ACC-030</i>	73
<i>Tabla 73: ACC-031</i>	74
<i>Tabla 74: ACC-032</i>	75
<i>Tabla 75: ACC-033</i>	76
<i>Tabla 76: ACC-034</i>	77
<i>Tabla 77: ACC-035</i>	78
<i>Tabla 78: ACC-036</i>	79
<i>Tabla 79: ACC-037</i>	80
<i>Tabla 80: ACC-038</i>	81
<i>Tabla 81: Ejemplo Tabla de Datos para Problemas</i>	92
<i>Tabla 82: Datos Problema 1</i>	95
<i>Tabla 83: Datos Problema 2</i>	99
<i>Tabla 84: Datos Problema 3</i>	102
<i>Tabla 85: Datos Problema 4</i>	106
<i>Tabla 86: Datos Problema 5</i>	109
<i>Tabla 87: Datos Problema 6</i>	113
<i>Tabla 88: Datos Problema 7</i>	117
<i>Tabla 89: Datos Problema 8</i>	121
<i>Tabla 90: Datos Problema 9</i>	125
<i>Tabla 91: Datos Problema 10</i>	130
<i>Tabla 92: Salarios Trabajadores</i>	135
<i>Tabla 93: Coste Personal</i>	136
<i>Tabla 94: Coste Materiales</i>	136
<i>Tabla 95: Gastos Indirectos</i>	137
<i>Tabla 96: Presupuesto</i>	137

ÍNDICE DE ILUSTRACIONES

<i>Ilustración 1: Planificación Automática</i>	<i>18</i>
<i>Ilustración 2: Jerarquía Object</i>	<i>34</i>
<i>Ilustración 3: Jerarquía Place</i>	<i>35</i>
<i>Ilustración 4: Jerarquía Misión</i>	<i>36</i>
<i>Ilustración 5: Objetos Corriente, contador y estado_final.....</i>	<i>37</i>
<i>Ilustración 6: Ciclo de Desarrollo en Espiral</i>	<i>133</i>
<i>Ilustración 7: Gantt Planificación Inicial</i>	<i>134</i>
<i>Ilustración 8: Gantt Planificación final</i>	<i>134</i>

1. Introducción

La inteligencia Artificial que se definía ya en 1956 como “La ciencia e ingeniería dedicadas a la creación de máquinas inteligentes” ha evolucionado mucho en los últimos años, siendo hoy en día una de las ramas más importantes de la ingeniería informática

La Inteligencia Artificial tiene una gran cantidad de campos de aplicación, por ejemplo, el razonamiento automático, la computación biológica, el control inteligente, la robótica etc. Además, esta ciencia estudia cómo resolver mediante distintas técnicas un gran número de problemas de una forma automatizada, como pueden ser, reconocimiento de patrones, creatividad artificial, visión artificial, procesamiento del lenguaje, planificación, etc. Este documento se centrará en algunas de las técnicas de planificación automática (usadas por ejemplo en campos tan interesantes como la robótica o el razonamiento automático).

La planificación es un proceso de reflexión que consiste en elegir y organizar acciones previendo sus resultados. Esta reflexión o deliberación tiene como objetivo lograr de la forma más eficaz posible algunas metas preestablecidas. La planificación automática es una rama de la Inteligencia Artificial que estudia computacionalmente este proceso de reflexión.

En este Trabajo Fin de Grado se va a usar la planificación automática para generar historias narrativas. Para llevar a cabo la planificación automática de tareas es preciso disponer de un planificador con la capacidad de generar un plan para alcanzar un objetivo, y además poder aportarle al planificador una representación de las metas a alcanzar, una representación de las acciones que se pueden realizar y una representación del estado inicial. Por lo tanto, las historias narrativas se van a generar gracias a la implementación de un dominio en el que se definirán los predicados que podrán aparecer en los distintos estados y las acciones permitidas para lograr generar los planes que lleven a las metas preestablecidas.

Las historias narrativas que se van a generar en este trabajo tienen que ver con la guerra civil que se simula en el videojuego Far Cry 4. Esta temática es idónea para aplicar la planificación automática de tareas, ya que como es costumbre en los videojuegos de este género los personajes están divididos en dos bandos bien definidos y la dinámica del videojuego es controlar los personajes de un bando y eliminar a los personajes del otro bando llevando a cabo diversas misiones. Lo que se propone en este trabajo es dada la temática, planificar la asignación de misiones del bando amistoso y gestionar también mediante planificación las distintas acciones que permiten completar las misiones, y así generar distintas narraciones que expongan varias maneras de llevar a cabo la resolución del videojuego.

En este documento se presenta el desarrollo del dominio de planificación. Lo primero que se expone es el estado del arte el cual consta de una breve introducción a la planificación automática y una justificación de la versión del lenguaje PDDL utilizado

para implementar el dominio. También en esta sección se expone el planificador Sayphi en el que ejecutaremos el dominio y, por último, se hace un breve repaso de los distintos trabajos similares.

A continuación, en un apartado, se definen los objetivos y motivaciones, y en otro los requisitos funcionales y no funcionales que debe cumplir el trabajo.

En el siguiente apartado expondremos en detalle el diseño del sistema, donde analizaremos cuidadosamente todas las decisiones tomadas que tienen que ver con el modelado del sistema, se explica la estructura de objetos utilizada, se analizan los predicados y las acciones del dominio, se detalla cómo son los problemas que resuelve el dominio implementado y se muestran las oraciones que representan a las acciones.

La siguiente sección se centra en la experimentación, en la cual se definen varios problemas, los cuales son ejecutados en el planificador Sayphi utilizando el dominio implementado, y se analizan los planes que se generan.

Seguidamente, se presenta en otro apartado la gestión del proyecto. En este apartado se presenta tanto la planificación inicial que se preveía seguir, como la planificación real que se ha seguido y también se hace una estimación de costes del proyecto. Por último en este apartado se presenta el análisis socio-económico realizado y el marco regulador que afecta al trabajo.

Finalmente, se exponen las conclusiones obtenidas tras haber finalizado el trabajo, y se enumeran algunas de las mejoras que se pueden desarrollar en trabajos futuros.

2. Estado del arte

2.1. Planificación automática de tareas

La planificación automática de tareas es una disciplina de la inteligencia artificial que consiste en dado un problema a resolver, generar un plan de acciones de manera automática que, siguiéndolo lleve a la resolución del problema planteado. El plan de acciones se genera gracias a la utilización de un planificador.

Los planificadores deben ser dotados de una representación de las metas que se quieren alcanzar, una representación de las acciones que se pueden realizar y una representación del estado inicial. Además, el planificador debe tener la capacidad de generar un plan para alcanzar el objetivo. Un plan no es más que una secuencia o lista de acciones, que llevan de un estado inicial a un estado final.

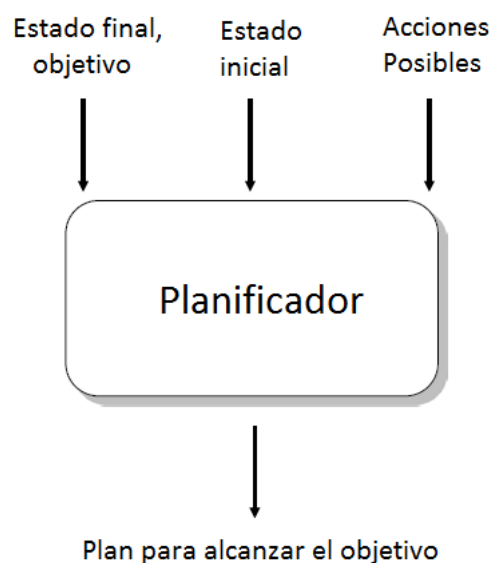


Ilustración 1: Planificación Automática

2.2. PDDL

El lenguaje de definición de dominios de Planificación (PDDL, Planinnig Domain Definition Language) es un intento de estandarizar los lenguajes de planificación. PDDL fue desarrollado en 1998 por Drew McDermott y varios amigos más [1]. Este lenguaje de planificación está inspirado en otros lenguajes de planificación como son STRIPS y ADL. PDDL fue desarrollado para ser usado en la competición internacional de planificación (IPC, International Planning Competition). Este lenguaje evolucionó en las siguientes competencias (IPC), generando así varias versiones del lenguaje PDDL, las cuales se describen a continuación.

- PDDL1.2. Este fue el lenguaje oficial para la primera y la segunda IPC celebradas en 1998 y 2000 respectivamente. En esta versión de PDDL ya se separaba el

modelo de planificación en dos partes, por un lado, quedaba el dominio del problema y por otro la descripción del problema.

- PDDL2.1: Esta versión de PDDL, fue el lenguaje oficial en el tercer IPC celebrada en 2002. Esta versión introduce funciones numéricas, métricas y acciones durativas.
- PDDL2.2: Esta versión fue el lenguaje oficial la cuarta IPC celebrada en 2004. A esta versión se añaden los predicados derivados, que permitían modelar la dependencia que existe entre hechos dados de otros hechos. También se añaden literales iniciales temporizados.
- PDDL3.0.: Esta versión fue el lenguaje oficial la quinta IPC celebrada en 2006. Como novedades esta versión presenta limitaciones del estado de trayectoria y preferencias para las metas de los problemas
- PDDL3.0.: Esta versión fue el lenguaje oficial la sexta y séptima IPC celebradas en 2008 y 2011 respectivamente. Esta es la última versión del lenguaje PDDL. En ella se añaden las funciones-objeto.

Para la realización de este trabajo se ha decidido usar PDDL1.2.

En PDDL1.2 el modelo de planificación queda separado en dominio y problema. A continuación, se describe la estructura que tiene cada parte del modelo.

Dominio

- Nombre: Lo primero que se especifica es el nombre del dominio.

(define (domain narrative)

- Características: Seguidamente se indican las características de PDDL que se van a utilizar en el dominio. Las características son unos requisitos que el planificador debe soportar para hacer frente al dominio. Estas características del dominio nos permiten imponer requisitos al planificador y al planificador le permiten descartar de forma rápida los problemas que no es capaz de afrontar. Las características que va a utilizar nuestro dominio son “strips”, “typing” y “equality”. Por defecto todo planificador debe soportar el modelo de STRIPS, por lo tanto, la característica de “strips” siempre se supone. La característica “typing” da soporte para tipar los objetos del dominio representando una jerarquía de tipos. Por último “equality” permite utilizar el predicado de “=” para la comparación de los valores de los objetos.

Para definir estas características primero se escribe la palabra “:requirements” para indicar que se van a empezar a especificar las características y luego se añaden las características oportunas precedidas del símbolo “:”, como se indica a continuación.

(:requirements :strips :typing :equality)

- Objetos: El siguiente paso es establecer la jerarquía de objetos.

```
(:types
  arm_ligera arm_pesada arm_explosiva - arm
  personaje arm construccion - object)
```

En el ejemplo que se muestra hay un tipo de objeto genérico “object” que da lugar a los subtipos “personaje”, “arm” y “construccion”, a su vez el objeto “arm” vuelve a desglosarse en otros tres subtipos de “arm”.

- Constantes: Una vez definida la jerarquía de objetos, podemos declarar las constantes, que son instancias de los objetos definidos en el paso anterior. Las constantes se definen en el dominio, por lo que todos los problemas que se creen las tendrán en común. La definición de constantes no es necesaria si el modelo no lo requiere.

```
(:constants
  zero one two three - contador
  politica religion - corriente )
```

- Predicados: Lo siguiente es definir los predicados que se utilizarán. Los predicados son usados para definir atributos de los objetos y para definir cualquier relación entre objetos. Para definir los predicados primero se escribe la palabra “:predicates” para indicar que se van a empezar a especificar los predicados, luego se crean los predicados que se consideren oportunos para el dominio, como se puede ver en el ejemplo que muestra a continuación.

```
(:predicates
  (personaje-vida ?personaje - personaje ?c - contador)
  (personaje-vida-quitada ?personaje - personaje ?c - contador)
  ...
)
```

La primera palabra representa el nombre del predicado, las palabras precedidas del símbolo “?” son variables y las palabras que se escriben delante de un guion “-” especifican el tipo de objeto de las variables que están detrás de ese guion “-”.

El predicado “personaje-vida” relaciona una variable de tipo “personaje” con una variable del tipo “contador”, esta relación es útil para especificar la vida que tiene un personaje.

- Acciones: Una vez definidos los predicados, lo siguiente es declarar las acciones, las cuales se dividen en cuatro partes.
 - Nombre: Lo primero es especificar el nombre de la acción.

```
(:action ataqueFusil
```

- Parámetros: los tipos de objetos de las variables que van a intervenir en la acción.

```
:parameters (?personaje1 ?personaje2 - personaje
              ?place - place ?arm - arm_ligera
              ?c1 ?c2 ?c3 - contador)
```

- Precondiciones: Predicados que deben cumplirse, es decir que son verdaderos en el estado en el que nos encontramos.

```
:precondition (and
               (at ?personaje1 ?place)
               (at ?personaje2 ?place)
               ...)
```

Por ejemplo, para esta acción debe cumplirse primeramente que en el estado haya un personaje en un lugar y luego que vuelva a haber un personaje en el mismo lugar, se especifica que el lugar tiene que ser el mismo pero el personaje puede ser repetido o no.

En las precondiciones también se pueden incluir predicados del tipo:

```
(not (= ?personaje1 ?personaje2))
```

Este predicado junto con los vistos anteriormente, obligan a que en el estado haya dos personajes distintos en el mismo lugar.

Otro tipo de precondiciones que podemos encontrar son del tipo:

```
(or
  (mision-descripcion ?ms2 ?const Destroyed)
  (mision-descripcion ?ms2 ?const Acquired)
)
```

Las cuales permiten seguir con la acción si una de las dos precondiciones que se proponen o las dos se cumplen.

- Efecto: Predicados que se crean o se borran cuando se ejecuta la acción. El efecto modifica el estado en el que nos encontrábamos antes de la acción.

```
:effect (and
         (not(personaje-vida-quitada ?personaje2 ?c2))
         (personaje-vida-quitada ?personaje2 ?c3))
```

Problema

- Nombre: Lo primero que hacemos es especificar el nombre del problema, igual que se hace en el dominio.

(define (problem p1)

- Domino: Lo siguiente que hay que hacer es indicar a qué dominio pertenece el problema.

(:domain narrative)

- Objetos: Una vez establecido el nombre del problema e indicado el domino al que pertenece, pasamos a crear instancias de los objetos, éstas instancias serán propias del problema, aparte el problema tendrá las instancias declaradas como constantes en el dominio.

(:objects

Banapur - place_senda

Varshakot Tirtha Shanath - place_const

Fortaleza_Yuma Fortaleza_Paul Fortaleza_Pagan - place_guardia

Sabal Ajay Amita - personaje_senda

Pagan Yuma Paul - personaje_guardia1

TenienteYuma TenientePaul TenientePagan - personaje_guardia2

...

)

- Estado inicial: Seguidamente haciendo uso de las instancias de los objetos creadas, establecemos el estado inicial en el que se encuentran las instancias de los objetos, relacionándolas gracias a los predicados creados en el dominio.

(:init

; Ajay

(at Ajay Banapur)

(personaje-corriente Ajay politica)

(personaje-corriente Ajay religiosa)

(have-an-obj Ajay AK-47)

(have-an-obj Ajay c4)

(personaje-vida Ajay two)

(personaje-vida-quitada Ajay zero)

...

)

- Objetivo: Por último, se especifica el objetivo final, que no es más que declarar predicados con unas instancias específicas. Estos predicados deben poder ser añadidos al estado del problema gracias a la ejecución de las acciones del dominio.

(:goal (and

(personaje-muerto Pagan)))

2.3. Sayphi

Sayphi es un planificador automático basado en heurísticas de búsqueda hacia delante como es FF. Además, Sayphi incluye técnicas de aprendizaje automático que permiten mejorar los resultados [2].

Sayphi permite ejecutar dominios con distintos algoritmos. A continuación, se hace una breve explicación de tres de ellos:

- Algoritmo Sayphi: Con este algoritmo se realiza una planificación totalmente determinista eligiendo en cada iteración la acción permitida más idónea para generar un plan.
- Algoritmo rrt: Permite añadir cierta aleatoriedad a la planificación.
- Algoritmo random-planner: Realiza una planificación totalmente aleatoria, genera un plan eligiendo en cada iteración una acción de entre todas las posibles de una forma aleatoria.

Además de permitir elegir el algoritmo con el que ejecutar el dominio en el planificador, Sayphi puede recibir distintos parámetros que pueden influir a la hora de generar los planes. Algunos de estos parámetros son los siguientes.

- Timeout: parámetro con el que se indica al planificador un tiempo máximo para generar el plan
- Number-plans: este parámetro indica el número de planes que se desean obtener.
- Iterations: indica el número máximo de acciones permitidas para generar el plan.
- probability-ehc: con este parámetro se indica la probabilidad de no realizar una acción aleatoria. Este parámetro es útil en el algoritmo rrt.

Entre las cualidades que tiene el planificador Sayphi, podemos destacar dos propiedades que hacen que sea un planificador idóneo para el problema que nos compete.

- Sayphi es independiente del dominio ya que su código no hace referencia a ninguno. Esto hace posible ejecutar cualquier dominio con Sayphi y generar planes.
- Sayphi gracias a las técnicas de búsqueda que tiene, puede generar planes de manera pseudo-aleatoria, escogiendo acciones determinadas por los mecanismos de planificación implementados o escogiendo acciones con cierta aleatoriedad. Esta aleatoriedad la podemos usar empleando el algoritmo rrt.

2.4. Trabajos similares

Por suerte en la actualidad existen bastantes estudios sobre la planificación automática aplicada a la generación de historias narrativas.

Un trabajo similar al presentado en este documento es “Narrative Generation in Entertainment: Using Artificial Intelligence Planning” de Richard A George [3]. El objetivo de este proyecto es la generación de diálogos entre distintos personajes. Los personajes de las historias que se generan en este proyecto presentan variables como género, edad, experiencias, personalidades, ambiciones, objetivos y diferentes variables de relación entre ellos. Para la implementación de este trabajo también se ha utilizado el lenguaje de programación PDDL. Este trabajo va un paso más allá y dota al usuario de una interfaz gráfica para crear los distintos personajes y además pretende que tras la generación de las historias, un software de procesamiento de lenguaje natural lea en voz alta los planes generados.

Otro trabajo con el que este proyecto comparte bastantes objetivos es “Story Generation in PDDL Using Character Moods: A Case Study on Iliad’s First Book” de Andrea Marrella y Stavros Vassos de la universidad Sapienza de Roma [4]. El objetivo de este trabajo es generar historias narrativas mediante planificación automática. El tema de las narrativas que se quieren generar tiene que ver con el libro de la Ilíada de Homero. Para la generación de las historias se ha implementado un dominio en PDDL. Como motor de la narración se han tomado la disposición que tienen unos personajes frente a otros.

Por último, hay dos trabajos bastantes similares a este, por un lado, esta “Generación Automática de Narrativas” de P. Pérez [5] y por otro “GENERACIÓN AUTOMÁTICA DE NARRATIVAS: Aventura Fantástica” de Francisco Javier Blanco Álvarez [6]. Ambos trabajos han sido realizados en la Universidad Carlos III de Madrid. Además, ambos trabajos se han implementado con el lenguaje PDDL1.2 y han sido probados en el planificador SAYPHI igual que el expuesto en este documento.

El trabajo de P.Pérez trata la generación automática de narrativas de género negro (mafia, asesinatos, policías...) y el trabajo de Francisco Javier Blanco Álvarez, trata de igual forma la generación automática de narrativas, pero el tema de las historias que genera su trabajo tiene que ver con la película de el Señor De Los Anillos.

En el presente documento se pretende realizar una aportación más a este problema. Como ya se ha comentado anteriormente las historias narrativas que se van a generar en este trabajo tienen que ver con la temática del juego Far Cry 4. Para generar las historias narrativas se va a aplicar la planificación tanto para la asignación de misiones como para la resolución de las mismas, abandonando la asignación de motivos finales a cada personaje tal y como se hace en los trabajos de P.Pérez y Francisco Javier Blanco Álvarez. Además, se pretende mejorar las narraciones solucionando algunos problemas como, por ejemplo, que un personaje de la narración realice un movimiento con un objetivo determinado y pudiendo realizar el objetivo realice otro

desplazamiento. Para ser más exactos en nuestras narrativas cuando un personaje se desplace a un lugar para realizar un ataque, el personaje no se moverá del lugar (quedará bloqueado) hasta que realice el ataque o hasta que no queden enemigos a los que atacar en el lugar.

En nuestras narrativas también se mejorará el sistema de salud de los personajes, representándolo mediante varios predicados que definiremos en el domino, creando un sistema de salud bastante similar a los de los videojuegos, en los que cada personaje aguantará un determinado número de ataques. Estas y otras ideas más son expuestas en el quito apartado de este documento, en el que se detalla el modelado del sistema que se ha realizado.

3. Objetivo y motivación

El objetivo central de este trabajo, es conseguir generar automáticamente narraciones coherentes y de interés, mediante la planificación automática. Para ello se genera un dominio, que, ejecutado en un planificador, sea capaz de generar un plan de resolución (historias) a distintos problemas. Además, se busca que los planes o historias encontradas tengan coherencia. Por último, se pretende que para un mismo problema se puedan encontrar un gran número de planes distintos, pudiendo incorporar cierta aleatoriedad al planificador en el que se ejecutará el dominio.

La temática de las historias que se pretenden generar tiene que ver con la guerra civil de la región que se simula en el juego Far Cry 4. Por lo tanto, el objetivo es narrar las acciones que van a ejecutar los personajes de cada Bando de la guerra. Dado que la región que se simula en el juego es bastante extensa, se van a generar narraciones que no incluirán todos los detalles que simula Ubisoft dentro del juego, pero se pretende que, aunque las historias que se generen en este trabajo no tengan todos los personajes, lugares, criaturas que aparecen en el juego, éstas sean lo bastante significativas como para sumergir al lector en la región que simula el videojuego Far Cry 4.

La motivación principal es conseguir que las historias generadas por el dominio implementado puedan ser utilizadas por distintas empresas relacionadas con la industria del entretenimiento como, por ejemplo:

- Canales de televisión: para generar sus propios programas, series o dibujos animados
- Productoras de cine: proporcionándoles un método automático para elaborar los guiones de sus películas.
- Empresas de videojuegos: las cuales podrían utilizar esta técnica para elaborar las historias de los videojuegos y dada la gran cantidad de historias diferentes sobre un mismo tema que se pueden generar podrían hacer única la experiencia de cada usuario, pudiendo llegar a dotar a los videojuegos de varios niveles de rejugabilidad.
- Editoriales: pudiendo usar el dominio implementado para elaborar narraciones de libros simples o comics.

4. Análisis del problema

A continuación se muestran una lista de los servicios que debe proporcionar el sistema (requisitos funcionales) y una lista de las restricciones que presenta el sistema (requisitos no funcionales).

4.1. Requisitos Funcionales

Nombre	Lenguaje de programación PDDL 1.2	Código	RF-001
Descripción	El dominio y los problemas, estarán escritos en el lenguaje de programación PDDL1.2		
Prioridad	Alta		

Tabla 1: RF-001

Nombre	Ejecución con dominio y problema.	Código	RF-002
Descripción	El planificador ejecutará un dominio y un problema escrito en el lenguaje de programación PDDL1.2		
Prioridad	Alta		

Tabla 2: RF-002

Nombre	Ejecución aleatoria	Código	RF-003
Descripción	El planificador podrá ejecutarse en modo aleatorio.		
Prioridad	Alta		

Tabla 3: RF-003

Nombre	Modificar aleatoriedad	Código	RF-004
Descripción	El usuario podrá modificar la probabilidad de que el sistema se ejecute en modo aleatorio.		
Prioridad	Baja		

Tabla 4: RF-004

Nombre	Modificar iteraciones	Código	RF-005
Descripción	El usuario podrá modificar el número de iteraciones del planificador.		
Prioridad	Baja		

Tabla 5: RF-005

Nombre	Ejecución mediante heurísticas	Código	RF-006
Descripción	El planificador se ejecutará mediante heurísticas.		
Prioridad	Alta		

Tabla 6: RF-006

Nombre	Creación de nuevos problemas	Código	RF-007
Descripción	El usuario podrá crear problemas nuevos para el dominio implementado.		
Prioridad	Alta		

Tabla 7: RF-007

Nombre	Elección de dominio a ejecutar	Código	RF-008
Descripción	El usuario podrá elegir qué dominio se ejecutará en el planificador.		
Prioridad	Alta		

Tabla 8: RF-008

Nombre	Elección de problema a ejecutar	Código	RF-009
Descripción	El usuario podrá elegir qué problema se ejecutará en el planificador.		
Prioridad	Alta		

Tabla 9: RF-009

Nombre	Mostrar solución	Código	RF-010
Descripción	El planificador mostrará la solución en la consola de comandos.		
Prioridad	Alta		

Tabla 10: RF-010

Nombre	Modificar tiempo de ejecución	Código	RF-011
Descripción	El usuario podrá determinar cuál es el tiempo máximo de ejecución de la planificación.		
Prioridad	Baja		

Tabla 11: RF-011

Nombre	Mostrar si hay solución	Código	RF-012
Descripción	El planificador mostrará por la consola de comandos si hay solución.		
Prioridad	Alta		

Tabla 12: RF-012

Nombre	Modificar las oraciones de salida	Código	RF-013
Descripción	El usuario podrá modificar las oraciones, que representan a cada acción del dominio implementado.		
Prioridad	Baja		

Tabla 13: RF-013

Nombre	Mostrar variaciones en los resultados.	Código	RF-014
Descripción	El planificador podrá mostrar cuánta variación existe entre dos planes que encuentre.		
Prioridad	Baja		

Tabla 14: RF-014

4.2. Requisitos No Funcionales

Nombre	Sistemas operativos	Código	RF-001
Descripción	El planificador se podrá ejecutar en Windows, Linux y Mac OS.		
Prioridad	Alta		

Tabla 15: RNF-001

Nombre	Tiempo Limite	Código	RF-002
Descripción	El sistema no podrá tardar más del tiempo límite establecido por el usuario.		
Prioridad	Alta		

Tabla 16: RNF-002

Nombre	SBCL	Código	RNF-003
Descripción	Para ejecutar el planificador es necesario iniciar SBCL.		
Prioridad	Alta		

Tabla 17: RNF-003

5. Diseño del sistema

Como se ha dicho anteriormente, el objetivo del trabajo es crear un sistema compuesto por un dominio capaz de resolver distintos problemas, generando planes de acciones, en cada una de las resoluciones de los diferentes problemas que se plantean. Luego llevando a cabo una traducción de las acciones a frases con sentido propio, conseguir que los planes tengan coherencia y despierten interés.

Este apartado constará primeramente de una breve explicación del tema que se quiere abordar para llevar a cabo la narración, luego se expondrá el planteamiento que se ha utilizado para conseguir la iteración de los personajes (las misiones que conseguirán los personajes para realizar unas acciones u otras). Seguidamente se hará un análisis de la estructura de objetos realizada y una descripción detallada de los predicados y acciones que componen el dominio. También en este apartado se hará una explicación de las distintas partes que componen el estado inicial de los problemas utilizados para realizar la experimentación. Por último, se presentará el significado narrativo que tiene cada acción.

5.1. Tema

Vamos a utilizar los personajes y algunas de las historias que aparecen en el juego Far cry 4.

Para poner en situación al lector, a continuación, se va a llevar a cabo un breve resumen de la temática principal del videojuego.

La trama del juego se centra en Kirat una región situada en algún lugar del Himalaya. El protagonista del juego Ajay Ghale va a Kirat para esparcir las cenizas de su madre, ya que este fue su último deseo. Una vez en Kirat Ajay se da cuenta que la región está dividida en dos facciones.

Por un lado, la Guardia Real liderada por Pagan Min, actual Dictador de la región, Pagan es apoyado por varios capitanes, Yuma y Paul, los cuales también tienen a su disposición fieles seguidores que están a su servicio. La Guardia Real tiene el dominio de Kirat.

Por otro lado, se encuentra la Senda Dorada la cual esta refugiada en Banapur una ciudad al sur de Kirat. La Senda Dorada está liderada por Amita y Sabal. Amita y Sabal quieren acabar con la dictadura de Pagan y comenzar a liderar Kirat. Por su parte Amita quiere acabar con las bases políticas de Pagan Min y adquirir ella el poder político. La lucha de Sabal se centra más en recuperar la religión tradicional de Kirat, liberar todas las estatuas, monasterios y templos que domina la Guardia Real y que están siendo desprestigiados.

Al principio de la trama, Ajay Ghale viendo las injusticias que se están produciendo en la región, y el sufrimiento que está generando la Guardia Real a muchos ciudadanos, decide unirse a la Senda Dorada y unirse a la lucha de Sabal y Amita. Durante el juego Sabal y Amita convencen a Ajay para realizar sus misiones, las primeras más enfocadas

a salvar la religión de Kirat y las otra dirigidas a la recuperación económica y política de Kirat. Pero ambos líderes tendrán en común el objetivo de liberar Kirat de la guardia Real.

5.2. Planteamiento

La idea principal es simular la guerra civil que nos encontramos en el juego Far Cry 4. En el cual nos encontramos una facción que lidera la región de Kirat bajo una dictadura, y otra facción que quiere liberar al pueblo de Kirat, acabando con la dictadura.

Para conseguir que los personajes realicen distintas tareas, hemos elegido dotarles de objetivos, estos objetivos pueden ser de varios tipos y son presentados a continuación.

Los líderes principales de la Guardia Real estarán situados a lo largo de toda la historia en un lugar concreto (sus respectivas fortalezas). Y tendrán a sus disposiciones un lacayo cada uno, con el que compartirán un objetivo.

Los objetivos de la Guardia Real serán muy simples, para cada líder conseguir un arma concreta que está situada en el mapa de Kirat. Esta arma es la única que puede derrotarle (conseguir su criptonita y con ella en su poder salvar su vida). Además, como ellos no van asumir riesgos mandan a sus lacayos a por el arma. Por lo tanto, comparten el objetivo. Los lacayos aparte de tener el objetivo de salvar a su líder también tienen siempre la posibilidad de buscar por el mapa a personajes de distinta facción para atacarles.

Los objetivos de la Senda Dorada son más complejos, los dos líderes (Sabal y Amita) tienen la misión de Liberar Kirat. Solo los personajes que tienen esta misión podrán asumir misiones relacionadas a ésta y mandar misiones también relacionadas con esta meta, a personajes de su misma facción. Los personajes de la Senda Dorada no podrán ir en busca de personajes de la facción contraria, pero si podrán atacarles si están en el mismo lugar. Las misiones que pueden asumir los personajes de esta facción pueden ser de dos tipos:

- Misiones que tengan que ver con recuperar o destruir una construcción que esté bajo dominio enemigo. Para ello habrá que ir a la construcción, acabar con los soldados que estén defendiendo la construcción y finalmente completar la misión.
- Misiones que tengan que ver con conseguir un arma y matar a un personaje.

Con estas misiones se pretende conseguir que los personajes realicen acciones de movimiento, acciones de disparar, matar, coger armas, dar armas, destruir y liberar construcciones...

5.3. Definición de tipos de objetos

La definición de la estructura de objetos tiene dos objetivos: por un lado, que estos objetos sean útiles y puedan ser utilizados correctamente para modelar los predicados del dominio, de tal forma que aporten información a los distintos estados por los que pasa un problema y puedan ser utilizados en las acciones definidas en el dominio. Por otro lado, la estructura de objetos debe estar muy jerarquizada, para permitir a las acciones utilizar predicados que consten de objetos con pocas instancias y así evitar instanciaciones innecesarias.

Las instanciaciones innecesarias se evitan jerarquizando la estructura de datos. La instanciación innecesaria y una manera de evitarla se explica en el siguiente ejemplo.

Ejemplo: Se quiere permitir que en el dominio se lance una acción cuando se cumpla la precondition de que en el estado hay un personaje de la Senda Dorada en un lugar propio de la Senda. Para resolver esta acción habría dos opciones claras de modelado.

1. Una opción sería relacionar mediante predicados, personajes con lugares, personajes con facciones y lugares con facciones y utilizar estos predicados como precondiciones, utilizando la facción de la Senda Dorada. Por lo cual habría dos tipos de parámetros para esta acción, personajes y lugares ya que la facción será la de la Senda Dorada. De esta forma, habría que instanciar los predicados de las precondiciones con todas las instancias de los parámetros para comprobar que las precondiciones de la acción se cumplen o no. Lo cual supondría gran tiempo de instanciación, si el número de instancias de cada parámetro es elevado, pudiendo haber muchos personajes y muchos lugares.
2. Sin embargo, la solución que se propone en este trabajo para lanzar acciones similares a estas (evitando instanciación innecesaria) es tener bien jerarquizados los objetos del dominio. Para este caso concreto habría que relacionar mediante un predicado los personajes con los lugares. Además, la estructura de objetos del dominio deberá estar muy jerarquizada dividiendo los personajes en dos tipos (personajes de la Senda Dorada y personajes de la Guardia Real) y los lugares en otros dos tipos (lugares propios de la Senda Dorada y lugares propios de la facción Guardia Real). Por lo tanto, para que esta acción se ejecute solo habrá que tener en las precondiciones de la acción el predicado comentado anteriormente que relaciona a los personajes con los lugares. Además, los dos tipos de parámetros que recibirá esta acción serán del tipo personajes de la Senda Dorada y lugares de la Senda Dorada. Por lo tanto, como los personajes de la Senda Dorada son un subconjunto de personajes y lugares de la Senda Dorada son un subconjunto de lugares, el número de instancias del predicado que se comprueba en la precondition será mucho menor.

A continuación, se explica cómo se han jerarquizado los objetos utilizados en el dominio implementado.

Jerarquía Object

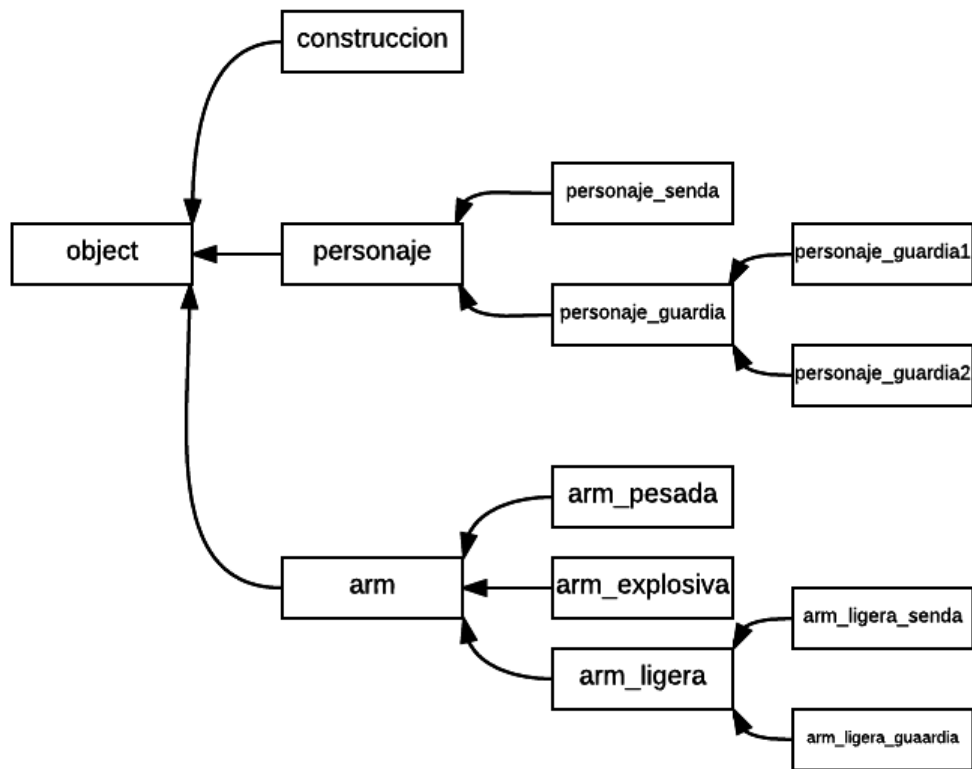


Ilustración 2: Jerarquía Object

El objeto principal de esta primera jerarquía es el objeto genérico object. Este tipo de objeto se divide en:

- **Construcción:** Las construcciones podrán estar en dos estados (destruidas o liberadas), o por el contrario en ninguno de estos estados. Además, las construcciones podrán tener armas.
Se considera que una construcción es accesible por un personaje de la Senda Dorada cuando se encuentra destruida o liberada. De distinta manera, se considera que una construcción es accesible por la Guardia Real cuando no se encuentra en ningún estado o está destruida.
Se dice que una construcción es accesible por un tipo de personaje para indicar que ese tipo de personaje tiene la posibilidad de coger el arma que se encuentra en la construcción.
- **Personajes:** En nuestro dominio, se van a considerar dos tipos de personajes, los personajes de la Senda Dorada y los personajes de la Guardia Real. Los personajes de la Guardia Real a su vez se dividen en dos clases de personajes. Personajes principales (personajes_guardia1), los cuales no podrán realizar movimientos, y además serán el objetivo de varias misiones de las historias, como ya veremos en las acciones y personajes secundarios

(personajes_guardia2) personajes irrelevantes, que sí se pueden desplazar de unos lugares a otros para lograr sus objetivos.

Los personajes, serán los principales tipos de objetos de las acciones de nuestro dominio, los personajes tendrán misiones y conseguirán nuevas misiones, se moverán por los distintos lugares, dispararán a otros personajes enemigos, destruirán o liberarán construcciones para conseguir armas poderosas, etc.

- Arma: Las armas son otro tipo de objeto importante en nuestro dominio. Los personajes y las construcciones tendrán en su poder las armas del juego.

Las armas se dividen en tres tipos:

- armas pesadas, armas que, en manos de unos personajes u otros, se utilizarán para matar o salvar a personajes de la Guardia Real principales. Los personajes de la Guardia Real solo podrán ser atacados con este tipo de armas.
- armas explosivas, las cuales las podrán utilizar los personajes de la Senda Dorada para acabar con todos los enemigos de una construcción y poder liberarla o destruirla.
- armas ligeras, que aparte de permitir a los personajes destruir o liberar construcciones serán necesarias para que los personajes puedan atacar a personajes del tipo contrario al suyo. Las armas ligeras son de dos tipos, las armas ligeras que utilizarán los personajes de la Senda y las armas ligeras que utilizarán los personajes de la Guardia.

Jerarquía Place

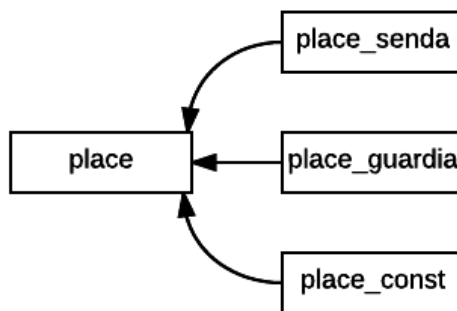


Ilustración 3: Jerarquía Place

En esta jerarquía, place es el objeto principal, y hace referencia a los lugares en los que se llevarán a cabo las tramas de las historias. Hay distintos tipos de lugares, lugares que serán propios de la senda Dorada, lugares de la Guardia Real y lugares en los que se encontrarán las construcciones.

Jerarquía Misión

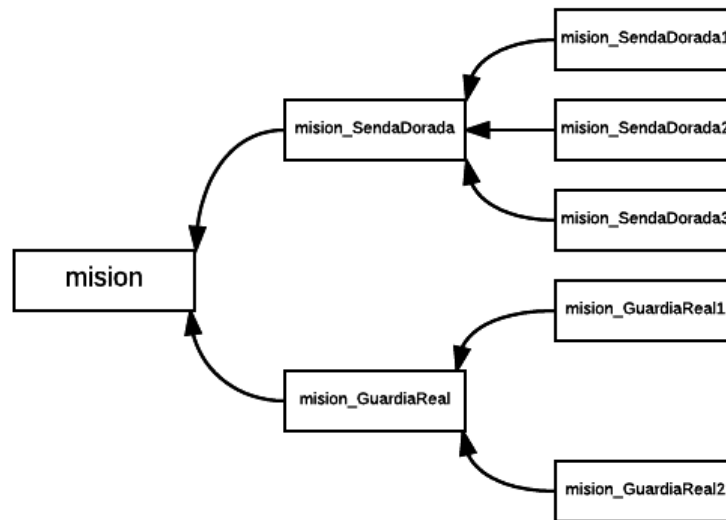


Ilustración 4: Jerarquía Misión

En la jerarquía de los objetos que están relacionados con las misiones, se puede observar que las misiones se dividen en misiones de la senda Dorada y misiones de la Guardia Real.

- Las misiones de la senda dorada son de tres tipos. El primer tipo será la misión principal de la Senda Dorada, que es liberar la región de Kirat a la cual la relacionaremos con los otros tipos de misiones, ya que para conseguir ésta no hay que hacer otra cosa que completar las otras. Las misiones del segundo tipo son las que están relacionadas con liberar y destruir construcciones. Por ultimo las misiones del tercer tipo serán misiones que tienen que ver con acabar con personaje principales de la facción contraria consiguiendo las armas que pueden matar a éstos.
- Por su lado las misiones de la Guardia Real pueden ser de dos tipos. El primer tipo de misión tiene que ver con la misión principal de la Guardia Real que es dominar la región de Kirat, que se podría definir de la misma forma que la misión principal de la Senda Dorada. Por otro lado, el segundo tipo de misiones son las que están relacionadas con salvar a los líderes o personajes principales de la Guardia Real, consiguiendo destruir las armas para las cuales son vulnerables.

Objetos corriente, contador y estado_final.

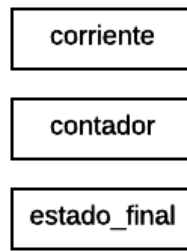


Ilustración 5: Objetos Corriente, contador y estado_final

Corriente se utilizará para definir a que corriente pertenece un personaje o una misión. En todos los problemas habrá dos tipos de corrientes (la corriente religiosa y la corriente política).

Contador, este objeto podrá tener el valor de zero, one, two o three y se utilizará para especificar en qué estado están algunos atributos de los personajes (vida) o construcciones (soldados).

Estado_final, para definir los estados en los que se encuentran algunas construcciones (destruida o liberada).

5.4. Definición de predicados

5.4.1. Predicado de Contadores

Como hemos visto en la definición de los tipos de objetos, hemos definido un objeto básico llamado contador. Las instancias de este objeto se definen en el dominio como constantes ya que serán comunes a todos los problemas que utilicemos y estas instancias serán el nombre de los números desde cero hasta tres escritos en inglés (zero, one, two, three). Este objeto se ha elaborado con la idea de llevar la cuenta de algunos atributos como la vida de los personajes o los soldados que hay en las construcciones. Pero este objeto no aporta ningún significado sin un predicado que ordene sus distintas instancias, un predicado que nos marque que “zero” es el primer contador, que “one” es el siguiente y después de “one” esta “two”, etc. Ya que si no sabemos cuál es el orden de sus instancias qué más da saber que la vida de un personaje es “two”, y que la vida que le han quitado a ese personaje es “one” si no está definido que “one” y “two” son consecutivos estos datos carecen de información. Por ello se ha modelado el predicado (next1 ?c1 - contador ?c2 - contador), con el que podemos definir en el estado inicial cuál es el ordenamiento de los contadores siendo ?c2 el contador que precederá a ?c1.

(next1 ?c1 - contador ?c2 - contador)

Nombre	next1	Código	PRE-001
Parámetros	?c1 - contador ?c2 - contador		
Descripción	Con este predicado conseguimos una numeración en orden de los números del 0-3. Y con esta numeración llevar la cuenta, por ejemplo, de la vida que tiene un personaje o los soldados que quedan en una construcción.		

Tabla 18: PRE-001

5.4.2. Predicados de Lugares

Para que las historias tengan mayor interés y coherencia, los personajes y construcciones, se situarán en determinados lugares. Para indicar esta propiedad se ha modelado el predicado “at” que relaciona un objeto “object” (personaje, construcción o arma) con un “place”.

Otro predicado que tiene que ver con los lugares es el predicado “numero-personajes-lugar-S”. Este predicado indica el número de personajes de la Senda Dorada que hay en un lugar. Cuando un personaje de la Guardia Real se mueve de un lugar a otro con el objetivo de atacar a un personaje de la Senda Dorada, el personaje de la Guardia Real quedará bloqueado en ese sitio hasta que realice un ataque o no pueda realizar un ataque. El predicado “numero-personajes-lugar-S” sirve para que el personaje de la Guardia Real que está bloqueado en un lugar determinado, pueda volver a poder moverse si es que los personajes de la Senda Dorada ya se han ido del lugar.

(at ?obj - object ?place - place)

Nombre	at	Código	PRE-002
Parámetros	?obj - object ?place - place		
Descripción	Se utiliza para ubicar un object en un lugar determinado.		

Tabla 19: PRE-002

(numero-personajes-lugar-S ?place - place ?contador - contador)

Nombre	numero-personajes-lugar-S	Código	PRE-003
Parámetros	?place - place ?contador - contador		
Descripción	Se utiliza para indicar el número de personajes de la Senda Dorada que hay en un lugar.		

Tabla 20: PRE-003

5.4.3. Predicados para Construcciones

Las construcciones, como ya se ha indicado, se sitúan en un lugar, y además tienen la posibilidad de tener un arma. Dicho arma en ocasiones puede ser deseada por un personaje, el cual dependiendo del tipo de personaje que sea, estando en el mismo lugar que la construcción podrá tener acceso al arma o no. Los personajes de la Senda Dorada tendrán acceso al arma de una construcción cuando ésta se encuentre liberada o destruida. Sin embargo, un personaje de la Guardia Real tendrá acceso al arma de la construcción, cuando no se encuentra en ningún estado, o está destruida. Para indicar el estado de una construcción se ha modelado el predicado “is”.

Dicho esto, es evidente que las construcciones se podrán liberar o destruir, y presumiblemente esta liberación y destrucción será llevada a cabo por personajes de la Senda Dorada que son los que necesitan que la construcción se encuentre en uno de estos estados. Para conseguir liberar o destruir una construcción, ésta no deberá tener soldados vivos custodiándola, como ya veremos en las acciones implementadas en él dominio.

Para modelar los soldados que custodian una construcción en cada momento, se ha relacionado a las construcciones con un contador que indicará cuántos soldados la custodian y de igual manera se las ha relacionado con otro contador, que indica cuántos soldados de los que están custodiando la construcción han sido eliminados. Para estas dos relaciones lógicas, se han modelado los predicados “construccion-soldados” y “construccion-soldados-muertos”.

(is ?obj - construccion ?st - estado)

Nombre	is	Código	PRE-004
Parámetros	?obj - construccion ?st - estado		
Descripción	Este predicado nos ayuda a declarar el estado de una construcción, ya que, en algunas ocasiones, las construcciones pasarán de no tener ningún estado, a ser liberadas o destruidas.		

Tabla 21: PRE-004

(construccion-soldados ?const - construccion ?c - contador)

Nombre	construccion-soldados	Código	PRE-005
Parámetros	?c1 - contador ?c2 - contador		
Descripción	Con este predicado definimos cuántos soldados tiene una determinada construcción.		

Tabla 22: PRE-005

(construccion-soldados-muertos ?const - construccion ?c - contador)

Nombre	construccion-soldados-muertos	Código	PRE-006
Parámetros	?const - construccion ?c - contador		
Descripción	Este predicado nos sirve para llevar la cuenta de los soldados pertenecientes a una construcción, que han sido eliminados.		

Tabla 23: PRE-006

5.4.4. Predicados para Personajes

Los personajes como elementos centrales de las narrativas, tendrán varios atributos y podrán estar en diferentes estados.

En cuanto a los atributos. Los personajes podrán ser de una corriente u otra, (Política y/o Religiosa). Este atributo será asignado a personajes del tipo Senda Dorada, ya que solo se utilizará este atributo en acciones propias de ellos. Este atributo se le podrá otorgar a un personaje haciendo uso del predicado “personaje-corriente”. Otros atributos importantes de los personajes son la vida que tienen y la vida que se les ha quitado, estos atributos se consiguen relacionando a los personajes con contadores, igual que se ha indicado en el caso de los soldados que custodian las construcciones. Estos atributos se han modelado con los predicados “personaje-vida” y “personaje-vida-quitada”.

Los personajes podrán encontrarse en unos determinados estados, como son el estado de muerto y ocupado. Estos estados se han modelado con los predicados “personaje-muerto” y “personaje-ocupado”. El predicado “personaje-ocupado” indica que al personaje se le ha asignado una misión. Por lo tanto, no se le podrá asignar otra misión hasta que no se complete la que tiene asignada.

Otro estado en el que se puede encontrar un personaje es el que se define en el predicado “tiene-que-disparar”. Un personaje está en este estado cuando se desplaza de un lugar a otro para atacar a un enemigo. Si un personaje está en este estado no podrá realizar ninguna acción de movimiento. Un personaje deja de estar en este estado cuando realiza un ataque o cuando quiere desplazarse a otro lugar y se encuentra en un sitio donde no puede realizar ataques porque no hay personajes a los que atacar.

Otros predicados que tienen que ver directamente con los personajes, son los predicados “criptonita” y “asesinado-por”. El predicado “criptonita” relaciona a un personaje con un arma poderosa que puede acabar con él, y el predicado “asesinado-por” relaciona dos personajes e indica que un personaje está muerto porque ha sido abatido por otro.

(personaje-corriente ?personaje - personaje ?corriente - corriente)

Nombre	personaje-corriente	Código	PRE-007
Parámetros	?personaje - personaje ?corriente - corriente		
Descripción	Sirve para asignar a un personaje una corriente.		

Tabla 24: PRE-007

(personaje-vida ?personaje - personaje ?c - contador)

Nombre	personaje-vida	Código	PRE-008
Parámetros	?personaje - personaje ?c - contador		
Descripción	Con este predicado definimos cuántos ataques tiene que recibir un personaje para ser eliminado.		

Tabla 25: PRE-008

(personaje-vida-quitada ?personaje - personaje ?c - contador)

Nombre	personaje-vida-quitada	Código	PRE-009
Parámetros	?personaje - personaje ?c - contador		
Descripción	Este predicado lleva la cuenta de los ataques que ha sufrido un personaje concreto.		

Tabla 26: PRE-009

(personaje-muerto ?personaje - personaje)

Nombre	personaje-muerto	Código	PRE-010
Parámetros	?personaje - personaje		
Descripción	Sirve para declarar que un personaje ha muerto.		

Tabla 27: PRE-010

(personaje-ocupado ?personaje - personaje)

Nombre	personaje-ocupado	Código	PRE-011
Parámetros	?personaje - personaje		
Descripción	Este predicado se utiliza para declarar que un personaje está ocupado. Este predicado nos será útil a la hora de implementar las acciones, pues en ellas, declararemos que un personaje queda ocupado cuando obtiene alguna de las misiones de la senda Dorada "mision_SendaDorada2" o "mision_SendaDorada3", y en estas acciones definiremos que un personaje ocupado no puede obtener otras misiones de este tipo		

Tabla 28: PRE-011

Nombre	Tiene-que-disparar	Código	PRE-012
Parámetros	?personaje - personaje		
Descripción	Este predicado se utiliza para declarar que un personaje está en el estado tiene que disparar. Este estado lo podrá abandonar cuando realice un ataque o no sea posible realizar un ataque.		

Tabla 29: PRE-012

(criptonita ?personaje - personaje ?arm - arm)

Nombre	criptonita	Código	PRE-013
Parámetros	?personaje - personaje ?arm - arm		
Descripción	Con este predicado asignamos que un personaje solo puede ser derrotado con un arma concreta.		

Tabla 30: PRE-013

(asesinado-por ?personaje - personaje ?personaje2 - personaje)

Nombre	asesinado-por	Código	PRE-014
Parámetros	?personaje1 - personaje ?personaje2 - personaje		
Descripción	Este predicado nos ayuda a guardar quien ha llevado a cabo la muerte de otro personaje. El personaje 1 es asesinado por el personaje2		

Tabla 31: PRE-014

5.4.5. Predicados para Armas

Las armas como ya se ha indicado varias veces podrán estar en poder de personajes y construcciones. Si un arma no pertenece ni a un personaje ni a una construcción, esa arma no podrá participar en las narrativas. Para indicar la posesión de armas de los personajes y construcciones se ha creado el predicado “have-an-obj”.

Además de poder tener armas, los personajes podrán desear tener armas que no tienen y deben tener si quieren completar determinadas misiones. El predicado que se ha modelado para indicar que un personaje quiere tener un arma es “wish-an-obj”.

(have-an-obj ?object - object ?arm - arm)

Nombre	have-an-obj	Código	PRE-015
Parámetros	?object - object ?arm - arm		
Descripción	Este predicado nos ayuda a saber qué armas tiene un personaje o una construcción.		

Tabla 32: PRE-015

(wish-an-obj ?personaje - personaje ?arm - arm)

Nombre	wish-an-obj	Código	PRE-016
Parámetros	?personaje - personaje ?arm - arm		
Descripción	Con este predicado declaramos que un personaje quiere un arma. Gracias a este predicado, en las acciones podemos definir que un personaje vaya en busca de un arma que quiere.		

Tabla 33: PRE-016

5.4.6. Predicados para Misiones

Los predicados modelados que tienen que ver con las misiones son:

- mision-bloqueada
- mision-complete
- mision-corriente
- have-mision
- mision-relaccionada
- misionS2-descripcion
- misionS3-descripcion
- misionG2-descripcion

“mision-bloqueada” y “mision-complete” indican los estados en los que pueden estar las misiones. Algunas misiones podrán estar en estado bloqueado indicando que esa

misión ya está siendo llevada a cabo por un personaje, por lo tanto, esa misión no se le podrá asignar a otro personaje.

Para indicar que un personaje tiene una misión se utilizara el predicado "have-mision".

Algunas misiones podrán ser de una corriente u otra (Política y/o Religiosa). Esto se indica haciendo uso del predicado "misión-corriente".

El predicado "mision-relaccionada" tiene gran importancia ya que con él podemos indicar que hay misiones que surgen de la necesidad de querer conseguir una misión principal. En los problemas que se plantean en este documento, las distintas misiones en las que se embarcan los personajes de la Senda Dorada, surgen de la necesidad de querer completar una misión principal que tendrán algunos personajes de la Senda Dorada (tener esta misión les otorga la cualidad de líderes). Los personajes que tengan esta misión principal serán capaces de asumir y mandar misiones relacionadas con la misión principal.

Por último, los predicados "misionS2-descripcion" "misionS3-descripcion" "misionG2-descripcion", surgen de la necesidad de tener que dar una descripción de los distintos objetivos que habrá que conseguir en cada una de los diferentes tipos de misiones.

(mision-bloqueada ?mision - mision)

Nombre	mision-bloqueada	Código	PRE-017
Parámetros	?mision - mision		
Descripción	<p>Una misión quedará bloqueada cuando se le asigne a un personaje. Si una misión está bloqueada no podrá ser cogida por otro personaje.</p> <p>Las misiones que pueden quedar bloqueadas son las del tipo "mision_SendaDorada2" y "mision_SendaDorada3", que son las que están relacionadas con destruir y liberar construcciones o matar personajes principales.</p>		

Tabla 34: PRE-017

(mision-complete ?personaje - personaje ?m - mision)

Nombre	mision-complete	Código	PRE-018
Parámetros	?personaje - personaje ?m - mision		
Descripción	Con este predicado dejamos registrado que un personaje ha completado una misión concreta.		

Tabla 35: PRE-018

(mision-corriente ?m - mision ?corriente - corriente)

Nombre	mision-corriente	Código	PRE-019
Parámetros	?m - mision ?corriente - corriente		
Descripción	Con este predicado definimos que una misión sea de una corriente. Las corrientes que habrá en el juego serán la política y la religiosa. Las misiones que estén relacionadas con una corriente solo podrán ser realizadas por personajes que sigan esa corriente, es decir que estén relacionados directamente con esa corriente mediante un predicado.		

Tabla 36: PRE-019

(have-mision ?personaje - personaje ?m - mision)

Nombre	have-mision	Código	PRE-020
Parámetros	?personaje - personaje ?m - mision		
Descripción	Este predicado sirve para saber que misión tiene un personaje. Relaciona a un personaje con una misión.		

Tabla 37: PRE-020

(mision-relacionada ?m - mision ?m2 - mision)

Nombre	mision-relacionada	Código	PRE-021
Parámetros	?m - mision ?m2 - mision		
Descripción	Algunas misiones están relacionadas con otras. Con este predicado dejamos de forma clara esta relación. Por ejemplo, la misión LiberarKirat en nuestros problemas está relacionada con LiberarConstrucciones MatarEnemigos. Con esta relación conseguimos que solo personajes que tengan la misión LiberarKirat puedan asumir misiones relacionadas a ésta o mandar misiones también relacionadas a otros personajes		

Tabla 38: PRE-021

(misionS2-descripcion ?m - mision_SendaDorada2 ?const - construccion ?estado - estado_final)

Nombre	misionS2-descripcion	Código	PRE-022
Parámetros	?m - mision_SendaDorada2 ?const - construccion ?estado - estado_final		
Descripción	Este predicado describe de una forma escueta lo que debe hacer un personaje que quiera completar una misión del tipo “mision_SendaDorada2”. Este tipo de misiones consisten en liberar o destruir una construcción. Por lo tanto, en la descripción de este tipo de misiones debe aparecer el nombre de la construcción y el estado en el que hay que dejarla, “liberada” o “destruida”.		

Tabla 39: PRE-022

(misionS3-descripcion ?m - mision_SendaDorada3 ?personaje - personaje)

Nombre	misionS3-descripcion	Código	PRE-023
Parámetros	?m - mision_SendaDorada3 ?personaje - personaje		
Descripción	Este predicado describe de una forma escueta lo que debe hacer un personaje que quiera completar una misión del tipo “mision_SendaDorada3”. Este tipo de misiones consisten en matar a un personaje. Por lo tanto, en la descripción solo aparece el personaje que debe ser abatido.		

Tabla 40: PRE-023

(misionG2-descripcion ?m - mision_GuardiaReal2 ?personaje - personaje)

Nombre	misionG2-descripcion	Código	PRE-024
Parámetros	?m - mision_GuardiaReal2 ?personaje - personaje		
Descripción	Este predicado describe de una forma escueta lo que debe hacer un personaje que quiera completar una misión del tipo “mision_GuardiaReal2”. Este tipo de misiones consisten en salvar a un personaje. Por lo tanto, en la descripción solo aparece el personaje que debe ser salvado.		

Tabla 41: PRE-024

5.5. Definición de acciones

La definición de acciones es la sección principal del dominio. En este apartado se procede a hacer una descripción detallada de cada una de las acciones implementadas en el dominio. Para llevar a cabo la descripción se va a utilizar una tabla con el siguiente formato:

Nombre		Código	
Descripción			
Parámetros			
Precondiciones			
Efectos			

Tabla 42: Tabla Para Acciones

Donde:

- Nombre: Especifica el nombre de la acción.
- Código: Cada acción será identificada con un código, con el siguiente formato ACC-XXX.
 - XXX: indica el número del código de la misión empezando en 001.
- Descripción: La descripción es un breve resumen de la funcionalidad de la acción.
- Parámetros: tipos de variables que intervienen en la acción.
- Precondiciones: Para que una acción se pueda ejecutar deberá cumplir las precondiciones. En este apartado se hará una breve explicación de los predicados que deben estar en el dominio en cada momento para que la acción cumpla sus precondiciones y pueda ser lanzada.
- Efectos: Explicación de las modificaciones que se producen en el estado cuando se ejecuta la acción.

Las diferentes acciones que han sido implementadas se pueden dividir en distintos bloques:

- Acciones para las asignaciones de misiones.
- Acciones para las asignaciones de las corrientes de los personajes.
- Acciones para permitir a los personajes conseguir armas.
- Acciones que permiten a los personajes atacar a las diferentes construcciones.
- Acciones que permiten a los personajes realizar movimientos.
- Acciones que permiten a los personajes atacarse con las armas.

5.5.1. Misiones

Nombre	mandar-mision1	Código	ACC-001
Descripción	Acción que se ejecuta para que un líder de la Senda Dorada mande una misión a un personaje de su misma facción y corriente, del tipo liberar o destruir un edificio.		
Parámetros	<p>?personaje1 ?personaje2 - personaje_senda</p> <p>?place - place</p> <p>?ms1 - mision_SendaDorada1</p> <p>?ms2 - mision_SendaDorada2</p> <p>?corriente - corriente</p>		
Precondiciones	<p>Tiene que haber en el estado, dos personajes de la Senda Dorada distintos, que no estén muertos y que sean de la misma corriente.</p> <p>El primer personaje tiene que tener una misión principal de tipo “mision_SendaDorada1”. El segundo personaje no tiene que tener una misión de ese tipo. Además, este personaje, no tiene que estar ocupado.</p> <p>También en el estado debe haber una misión del tipo “mision_SendaDorada2”, relacionada con la misión principal que tiene el primer personaje, y que además la misión esté relacionada con la corriente que comparten ambos personajes.</p> <p>Por último, se debe cumplir que la misión encontrada, no esté bloqueada.</p>		
Efectos	El personaje que no tiene la misión principal de la Senda Dorada, consigue la misión del tipo “mision_SendaDorada2”, la misión queda bloqueada y el personaje ocupado.		

Tabla 43: ACC-001

Nombre	mandar-mision2	Código	ACC-002
Descripción	Acción que se ejecuta para que un líder de la Senda Dorada mande una misión a un personaje de su misma facción y corriente, del tipo matar a un personaje principal de la otra facción.		
Parámetros	?personaje1 ?personaje2 - personaje_senda ?place - place ?ms1 - mision_SendaDorada1 ?ms3 - mision_SendaDorada3 ?corriente - corriente		
Precondiciones	<p>Tiene que haber en el estado dos personajes de la Senda Dorada distintos, que no estén muertos y que sean de la misma corriente.</p> <p>El primer personaje tiene que tener una misión principal de tipo “mision_SendaDorada1”. El segundo personaje no tiene que tener la misión de ese tipo. Además, este personaje, no debe estar ocupado.</p> <p>También en el estado debe haber una misión del tipo “mision_SendaDorada3”, relacionada con la misión principal que tiene el primer personaje, y que además la misión esté relacionada con la corriente que comparten ambos personajes.</p> <p>Por último, se debe cumplir que la misión encontrada no esté bloqueada.</p>		
Efectos	El personaje que no tiene la misión principal de la Senda Dorada consigue la misión del tipo “mision_SendaDorada3”. La misión queda bloqueada y el personaje ocupado.		

Tabla 44: ACC-002

Nombre	aprender-mision1	Código	ACC-003
Descripción	Un líder de la Senda Dorada asume una misión relacionada con la misión principal de la Senda Dorada, la cual le hace líder. La misión que asume es del tipo destruir o adquirir una construcción.		
Parámetros	?personaje - personaje_senda ?ms1 - mision_SendaDorada1 ?ms2 - mision_SendaDorada2 ?corriente - corriente		
Precondiciones	<p>Para esta misión debe haber en el estado, un personaje de la Senda Dorada vivo, no ocupado y con la misión principal de la Senda Dorada (una misión del tipo “mision_SendaDorada1”).</p> <p>También en el estado tiene que haber una misión del tipo “mision_SendaDorada2”, relacionada con la misión principal que tiene el personaje, y que además la misión esté relacionada con la misma corriente a la que pertenece el personaje.</p> <p>Para acabar la misión del tipo “mision_SendaDorada2” no debe estar bloqueada.</p>		
Efectos	El personaje consigue una misión del tipo “mision_SendaDorada2”, la misión queda bloqueada y el personaje ocupado.		

Tabla 45: ACC-003

Nombre	aprender-mision2	Código	ACC-004
Descripción	Un líder de la Senda Dorada asume una misión relacionada con la misión principal de la Senda Dorada, la cual le hace líder. La misión que asume es del tipo matar a un personaje principal de la otra facción.		
Parámetros	?personaje - personaje_senda ?ms1 - mision_SendaDorada1 ?ms3 - mision_SendaDorada3		
Precondiciones	<p>Para esta misión debe haber en el estado un personaje de la Senda Dorada vivo, no ocupado y con la misión principal de la Senda Dorada (una misión del tipo "mision_SendaDorada1").</p> <p>También en el estado tiene que haber una misión del tipo "mision_SendaDorada3", relacionada con la misión principal que tiene el personaje.</p> <p>Para acabar la misión del tipo "mision_SendaDorada3" no debe estar bloqueada.</p>		
Efectos	El personaje consigue una misión del tipo "mision_SendaDorada3", la misión queda bloqueada y el personaje ocupado.		

Tabla 46: ACC-004

Nombre	aprender-misionSenda-principal	Código	ACC-005
Descripción	<p>Acción para que un personaje que no tenga la misión principal de la Senda Dorada (misión del tipo "mision_SendaDorada1") pueda conseguirla.</p> <p>Esta acción se ha hecho porque en los problemas con los que se ha realizado la experimentación, únicamente tienen la misión principal dos personajes, que son considerados los lideres, y con esta misión ellos mandan y asumen misiones de otro tipo. Entonces si estos personajes mueren nadie tiene la misión principal y nadie manda ni asume misiones.</p>		
Parámetros	?personaje1 ?personaje2 ?personaje3 - personaje_senda ?ms1 - mision_SendaDorada1		
Precondiciones	<p>Tiene que haber dos personajes de la Senda Dorada muertos y distintos.</p> <p>De los dos personajes, uno tiene que ser de la corriente religiosa y otro de la corriente política.</p> <p>Ambos personajes tienen que tener la misma misión, del tipo "mision_SendaDorada1". Esta misión es la misión principal de la Senda Dorada, la misión de LiberarKirat.</p> <p>Tiene que haber un personaje de la Senda Dorada vivo que sea de la corriente de la Religion y de la Politica, y que no sea igual a ninguno de los dos personajes del principio. Además, este personaje no tiene que tener la misión, del tipo "mision_SendaDorada1", que compartían los otros personajes.</p>		
Efectos	Como resultado el personaje vivo consigue la misión principal de la Senda Dorada, del tipo "mision_SendaDorada1"		

Tabla 47: ACC-005

5.5.2. Corrientes

Nombre	aprender-corriente1	Código	ACC-006
Descripción	Un personaje que tiene como objetivo liberar Kirat adquiere de un compañero muerto, con su mismo objetivo, la corriente política.		
Parámetros	?personaje1 ?personaje2 - personaje_senda ?ms1 - mision_SendaDorada1		
Precondiciones	<p>Hay dos personajes de la Senda Dorada, uno muerto y otro vivo. Ambos comparten que los dos tienen una misión del tipo “mision_SendaDorada1”.</p> <p>El personaje muerto sigue la corriente Política de la Senda Dorada. Por el contrario, el personaje vivo no la sigue.</p>		
Efectos	Como resultado de esta acción el personaje vivo, empieza a seguir la corriente de su compañero muerto.		

Tabla 48: ACC-006

Nombre	aprender-corriente2	Código	ACC-007
Descripción	Un personaje que tiene como objetivo liberar Kirat adquiere de un compañero muerto, con su mismo objetivo, la corriente religiosa.		
Parámetros	?personaje1 ?personaje2 - personaje_senda ?ms1 - mision_SendaDorada1		
Precondiciones	<p>Hay dos personajes de la Senda Dorada, uno muerto y otro vivo, ambos comparten que los dos tienen una misión del tipo “mision_SendaDorada1”.</p> <p>El personaje muerto sigue la corriente Religiosa de la Senda Dorada. Por el contrario, el personaje vivo no la sigue.</p>		
Efectos	Como resultado de esta acción el personaje vivo empieza a seguir la corriente de su compañero muerto.		

Tabla 49: ACC-007

5.5.3. Conseguir Armas

Nombre	querer-objeto-por-mision-senda	Código	ACC-008
Descripción	Un personaje de la Senda Dorada empieza a querer un arma para poder completar su misión.		
Parámetros	?personaje1 - personaje_senda ?personaje2 - personaje_guardia1 ?ms3 - mision_SendaDorada3 ?arm - arm_pesada		
Precondiciones	<p>En el estado hay un personaje de la Senda Dorada vivo, el personaje tiene una misión del tipo "mision_SendaDorada3" la cual consiste en matar a un personaje.</p> <p>La misión tiene que tener una descripción en la cual se indique el nombre de un personaje principal de la Guardia Real vivo, que tiene que ser abatido. El personaje debe tener una criptonita, que será un arma pesada.</p> <p>El personaje de la Senda Dorada no debe tener el arma, ni tiene que quererla.</p>		
Efectos	Como resultado de la acción, el personaje de la Senda Dorada quiere esa arma para completar su misión.		

Tabla 50: ACC-008

Nombre	querer-objeto-por-mision-guardia1	Código	ACC-009
Descripción	Un personaje principal de la Guardia Real empieza a querer un arma para poder completar su misión.		
Parámetros	?personaje1 - personaje_guardia1 ?personaje2 - personaje_guardia1 ?mg2 - mision_GuardiaReal2 ?arm - arm_pesada		
Precondiciones	<p>En el estado hay un personaje principal de la Guardia Real vivo, el personaje tiene una misión del tipo “mision_GuardiaReal2”, la cual consiste en salvar a un personaje principal de la Guardia Real.</p> <p>La misión tiene que tener una descripción en la cual se indique el nombre de un personaje principal de la Guardia Real vivo, que tiene que ser salvado. Dicho personaje tendrá una criptonita, que será un arma pesada.</p> <p>El personaje que tiene la misión no debe tener el arma, ni tiene que quererla.</p>		
Efectos	Como resultado de la acción, el personaje que tiene la misión quiere el arma para completar su misión.		

Tabla 51: ACC-009

Nombre	querer-objeto-por-mision-guardia2	Código	ACC-010
Descripción	Un personaje secundario de la Guardia Real empieza a querer un arma para poder completar su misión.		
Parámetros	?personaje1 - personaje_guardia2 ?personaje2 - personaje_guardia1 ?mg2 - mision_GuardiaReal2 ?arm - arm_pesada		
Precondiciones	<p>En el estado hay un personaje secundario de la Guardia Real vivo, el personaje tiene una misión del tipo “mision_GuardiaReal2”, la cual consiste en salvar a un personaje principal de la Guardia Real.</p> <p>La misión tiene que tener una descripción, en la cual se indique el nombre de un personaje principal de la Guardia Real vivo, que tiene que ser salvado. Dicho personaje tendrá una criptonita, que será un arma pesada.</p> <p>El personaje que tiene la misión no debe tener el arma, ni tiene que quererla.</p>		
Efectos	Como resultado de la acción, el personaje que tiene la misión quiere el arma para completar su misión.		

Tabla 52: ACC-010

Nombre	coger-objeto-Senda	Código	ACC-011
Descripción	Un personaje de la senda dorada coge un arma que se encuentra en una construcción liberada o destruida.		
Parámetros	?personaje - personaje_senda ?arm - arm ?place - place ?const - construccion ?estado - estado_final		
Precondiciones	Hay un personaje de la Senda Dorada que no está muerto. El personaje se encuentra en un lugar, en el cual hay una construcción que tiene un arma. La construcción se encuentra destruida o adquirida.		
Efectos	El efecto es que el arma deja de estar en la construcción y pasa a pertenecer al personaje.		

Tabla 53: ACC-011

Nombre	coger-objeto-Guardia	Código	ACC-012
Descripción	Un personaje de la Guardia Real no principal coge un arma, que está en una construcción de la Guardia Real no liberada.		
Parámetros	?personaje - personaje_guardia2 ?arm - arm ?place - place ?const - construccion ?estado - estado_final		
Precondiciones	Hay un personaje de la Guardia Real que no está muerto. El personaje se encuentra en un lugar, en el cual hay una construcción que tiene un arma. La construcción se encuentra destruida o adquirida.		
Efectos	El efecto es que el arma deja de estar en la construcción y pasa a pertenecer al personaje.		

Tabla 54: ACC-012

Nombre	dar-objeto-Senda	Código	ACC-013
Descripción	Un personaje de la Senda Dorada le da un arma que no quiere a otro personaje también de la Senda Dorada, que quiere el arma.		
Parámetros	?personaje ?personaje2 - personaje_senda ?arm - arm_pesada ?place - place		
Precondiciones	Hay dos personajes de la Senda Dorada vivos y distintos. Además, están en el mismo lugar. Uno de los personajes tiene un arma que no necesita. El otro personaje quiere el arma porque le hace falta.		
Efectos	Como efecto de esta acción el personaje que tenía el arma deja de tenerla y el personaje que la quería consigue el arma.		

Tabla 55: ACC-013

Nombre	dar-objeto-Guardia1	Código	ACC-014
Descripción	Un personaje de la Guardia Real no principal le da un arma a otro personaje de la Guardia Real principal, que quiere dicha arma.		
Parámetros	personaje - personaje_guardia2 ?personaje2 - personaje_guardia1 ?arm - arm_pesada ?place - place		
Precondiciones	Hay dos personajes de la Guardia Real vivos y distintos, uno será principal y el otro no. Además, están en el mismo lugar". El personaje de la Guardia Real que no es principal tiene un arma, la cual puede quererla o no, no especificamos nada. El otro personaje quiere el arma.		
Efectos	Como efecto de esta acción el personaje que tiene el arma deja de tenerla y el personaje que la quería consigue tenerla.		

Tabla 56: ACC-014

Nombre	dar-objeto-Guardia2	Código	ACC-015
Descripción	Un personaje de la Guardia Real no principal le da un arma a otro personaje también de la Guardia Real no principal, que quiere dicha arma.		
Parámetros	?personaje - personaje_guardia2 ?personaje2 - personaje_guardia2 ?arm - arm_pesada ?place - place		
Precondiciones	Hay dos personajes de la Guardia Real no principales vivos y distintos. Además, están en el mismo lugar". Uno de los personajes tiene un arma, la cual puede quererla o no, no especificamos nada. El otro personaje quiere el arma		
Efectos	Como efecto de esta acción el personaje que tiene el arma deja de tenerla y el personaje que la quería consigue tenerla.		

Tabla 57: ACC-015

Nombre	coger-objeto-de-personaje-muerto	Código	ACC-016
Descripción	Un personaje coge un arma que tiene otro personaje que está muerto, y se encuentra en su mismo lugar.		
Parámetros	?personaje ?personaje2 - personaje ?arm - arm_pesada ?place - place		
Precondiciones	Hay un personaje que no está muerto, y otro personaje distinto que sí está muerto y que tiene un arma pesada en su poder. Ambos personajes se encuentran en el mismo lugar.		
Efectos	El efecto es que el arma deja de pertenecer al personaje muerto y pasa a pertenecer al personaje vivo.		

Tabla 58: ACC-016

Nombre	destruir-criptonita	Código	ACC-017
Descripción	Un personaje de la Guardia Real principal tiene en su poder el arma que puede acabar con él, y la destruye.		
Parámetros	?personaje - personaje_guardia1 ?arm - arm_pesada ?mg2 - mision_GuardiaReal2		
Precondiciones	Un personaje principal de la Guardia Real vivo tiene como misión salvarse y además tiene en su poder el único arma con el que un enemigo podría derrotarle.		
Efectos	El efecto es que el arma deja de estar en posesión de nadie, deja de estar en Kirat y el personaje completa su misión.		

Tabla 59: ACC-017

5.5.4. Ataques a construcciones

Nombre	ataqueSendaArma CortaEdificio	Código	ACC-018
Descripción	Un personaje elimina a un soldado de una construcción.		
Parámetros	?personaje - personaje_senda ?const - construccion ?place - place_const ?arm - arm_ligera ?c1 ?c2 ?c3 - contador		
Precondiciones	<p>Para que esta acción se ejecute, en el estado debe haber un personaje de la Senda Dorada vivo, el personaje debe estar en el mismo lugar que una construcción.</p> <p>La construcción no debe estar ni destruida ni liberada. Además, el personaje tiene que tener un arma ligera.</p> <p>El número de soldados de la construcción y el número de soldados muertos de esa construcción no debe ser igual.</p>		
Efectos	Como efecto, esta acción modifica el contador de soldados muertos de la construcción al contador siguiente.		

Tabla 60: ACC-018

Nombre	ataqueSenda ExplosivosEdificio	Código	ACC-019
Descripción	Un personaje elimina a todos los soldados que quedan en una construcción.		
Parámetros	?personaje - personaje_senda ?const – construccion ?place - place_const ?arm - arm_explosiva ?c1 ?c2 - contador		
Precondiciones	<p>Para que esta acción se ejecute, en el estado debe haber un personaje de la Senda Dorada vivo. El personaje debe estar en el mismo lugar que una construcción.</p> <p>La construcción no debe estar ni destruida ni liberada. Además, el personaje tiene que tener un arma explosiva.</p> <p>El número de soldados de la construcción y el número de soldados muertos de esa construcción no debe ser igual.</p>		
Efectos	Esta acción modifica el contador de soldados muertos de la construcción, al contador de soldados que hay en la construcción.		

Tabla 61: ACC-019

Nombre	destruir-construccion1	Código	ACC-020
Descripción	Un personaje destruye o libera una construcción. Además, ningún personaje tenía la misión de destruir o liberar dicha construcción.		
Parámetros	?personaje - personaje_senda ?ms2 - mision_SendaDorada2 ?const - construccion ?place - place_const ?c - contador ?estado - estado_final		
Precondiciones	<p>Para que se ejecute esta acción debe haber un personaje vivo de la Senda Dorada en el mismo lugar donde se encuentra una construcción.</p> <p>La construcción debe aparecer en la descripción de una misión que no esté bloqueada. La construcción no debe estar ni destruida ni liberada.</p> <p>El número de soldados de la construcción y el número de soldados muertos de esa construcción debe ser igual.</p>		
Efectos	<p>Como efecto de esta acción, la construcción pasa a estar en estado Destruída o Liberada según marque la misión.</p> <p>La misión que tenía como objetivo liberar o destruir a la construcción, se completa y se bloquea.</p>		

Tabla 62: ACC-020

Nombre	destruir-construccion2	Código	ACC-021
Descripción	Un personaje destruye o libera una construcción. Además, el personaje tiene la misión de destruir o liberar dicha construcción.		
Parámetros	?personaje - personaje_senda ?ms2 - mision_SendaDorada2 ?const - construccion ?place - place_const ?c - contador ?estado - estado_final		
Precondiciones	<p>Para que se ejecute esta acción debe haber un personaje vivo de la Senda Dorada en el mismo lugar donde se encuentra una construcción.</p> <p>La construcción debe aparecer en la descripción de una misión que esté bloqueada. El personaje además tiene esa misión.</p> <p>La construcción no debe estar ni destruida ni liberada.</p> <p>Por último, el número de soldados de la construcción y el número de soldados muertos de esa construcción debe ser igual.</p>		
Efectos	<p>Como efecto de esta acción, la construcción pasa a estar en estado Destruída o Liberada según marque la misión.</p> <p>La misión que tenía el personaje se completa, y éste deja de tener la misión y deja de estar ocupado.</p>		

Tabla 63: ACC-021

Nombre	destruir-construccion3	Código	ACC-022
Descripción	Un personaje destruye o libera una construcción. Además, otro personaje tenía la misión de destruir o liberar dicha construcción		
Parámetros	?personaje1 ?personaje2 - personaje_senda ?ms2 - mision_SendaDorada2 ?const - construccion ?place - place_const ?c - contador ?estado - estado_final		
Precondiciones	<p>Para que se ejecute esta acción debe haber un personaje vivo de la Senda Dorada en el mismo lugar donde se encuentra una construcción.</p> <p>La construcción debe aparecer en la descripción de una misión que esté bloqueada. Tiene que haber otro personaje distinto al anterior también vivo que esté ocupado en completar la misión que tiene que ver con la construcción.</p> <p>La construcción no debe estar ni destruida ni liberada.</p> <p>Por último, el número de soldados de la construcción y el número de soldados muertos de esa construcción debe ser igual.</p>		
Efectos	<p>Como efecto de esta acción, la construcción pasa a estar en estado Destruída o Liberada según marque la misión.</p> <p>La misión que tenía el personaje se completa, y éste deja de tener la misión y deja de estar ocupado.</p>		

Tabla 64: ACC-022

5.5.5. Movimientos

Nombre	ir-a-1	Código	ACC-023
Descripción	Mover a un personaje de la Senda Dorada, que tiene como misión destruir o adquirir una construcción, al lugar donde se encuentra la construcción.		
Parámetros	?personaje - personaje_senda ?ms2 - mision_SendaDorada2 ?const - construccion ?place – place ?place2 – place ?estado - estado_final ?c1 ?c2 ?c3 ?c4 - contador		
Precondiciones	Hay un personaje de la Senda Dorada vivo con una misión, en cuya descripción hay una construcción que debe ser destruida o adquirida. El personaje no está en el lugar donde se encuentra la construcción.		
Efectos	Como efecto se tiene que el personaje abandona el lugar donde se encuentra y va al lugar donde está la construcción. Además, en el lugar de salida habrá un personaje menos de la Senda Dorada y en el lugar de destino habrá un personaje más.		

Tabla 65: ACC-023

Nombre	ir-a-2-1	Código	ACC-024
Descripción	Mover a un personaje de la Senda Dorada que quiere un arma, al lugar donde se encuentra el arma. El arma debe estar en una construcción.		
Parámetros	?personaje - personaje_senda ?const - construccion ?place – place ?place2 - place ?arm - arm_pesada ?c1 ?c2 ?c3 ?c4 - contador		
Precondiciones	Hay un personaje de la Senda Dorada vivo, que quiere un arma. El arma se encuentra en una construcción. El personaje se encuentra en un lugar distinto al de la construcción donde se encuentra.		
Efectos	El efecto de esta acción es que el personaje deja de estar en su sitio y pasa a estar en el lugar donde se encuentra la construcción en la que está el arma. Además, en el lugar de salida habrá un personaje menos de la Senda Dorada y en el lugar de destino habrá un personaje más.		

Tabla 66: ACC-024

Nombre	ir-a-2-2	Código	ACC-025
Descripción	Mover a un personaje de la Guardia Real secundario que quiere un arma, al lugar donde se encuentra el arma. El arma debe estar en una construcción.		
Parámetros	?personaje - personaje_guardia2 ?const - construccion ?place – place ?place2 - place ¿arm - arm_pesada		
Precondiciones	Hay un personaje no principal de la Guardia Real vivo, que quiere un arma. El arma se encuentra en una construcción. El personaje se encuentra en un lugar distinto al de la construcción donde se encuentra. El personaje o no tiene que disparar o sí tiene que disparar, pero en el lugar donde se encuentra no quedan enemigos.		
Efectos	El efecto de esta acción es que el personaje deja de estar en su sitio y pasa a estar en el lugar donde se encuentra la construcción en la que está el arma. Si el personaje tenía que disparar, (y se ha movido porque no quedan enemigos) deja de tener el deber de disparar.		

Tabla 67: ACC-025

Nombre	Ir-a-3	Código	ACC-026
Descripción	Un personaje de la Senda Dorada tiene un arma que no quiere (no le hace falta), y además este arma la quiere otro personaje también de la Senda Dorada, por lo que movemos al personaje que tiene el arma hacia el lugar donde se encuentra el personaje que la quiere.		
Parámetros	?personaje ?personaje2 - personaje_senda ?arm - arm_pesada ?place ?place2 - place ?c1 ?c2 ?c3 ?c4 - contador		
Precondiciones	En el estado debe haber dos personajes de la Senda Dorada vivos. Uno de los personajes con un arma que no necesita, el otro con la necesidad de tener esa arma que tiene su compañero. Además, los personajes deben estar en sitios distintos.		
Efectos	El efecto de esta acción es que el personaje que tiene el arma deja de estar en su lugar y se mueve al lugar del personaje que quiere el arma. Además, en el lugar de salida habrá un personaje menos de la Senda Dorada y en el lugar de destino habrá un personaje más.		

Tabla 68: ACC-026

Nombre	ir-a-4-1	Código	ACC-027
Descripción	Un personaje de la Guardia Real no principal se mueve al lugar donde se encuentra otro personaje de la Guardia Real no principal. El movimiento se realiza para poder compartir un arma.		
Parámetros	?personaje - personaje_guardia2 ?personaje2 - personaje_guardia2 ?arm - arm_pesada ?place ?place2 - place		
Precondiciones	<p>En el estado debe haber dos personajes de la Guardia Real no principales vivos. Uno de los personajes con un arma, el otro con la necesidad de tener esa arma que tiene su compañero. Además, los personajes deben estar en sitios distintos.</p> <p>El personaje que tiene el arma o no tiene que disparar o sí tiene que disparar, pero en el lugar donde se encuentra no quedan personajes de la Senda Dorada.</p>		
Efectos	<p>El efecto de esta acción es que el personaje que tiene el arma deja de estar en su lugar y se mueve al lugar del personaje que quiere el arma.</p> <p>Si el personaje tenía que disparar, (y se ha movido porque no quedan enemigos) deja de tener el deber de disparar.</p>		

Tabla 69: ACC-027

Nombre	ir-a-4-2	Código	ACC-028
Descripción	Un personaje de la Guardia Real no principal se mueve al lugar donde se encuentra un personaje de la Guardia Real principal. El movimiento se realiza para poder compartir un arma.		
Parámetros	?personaje - personaje_guardia2 ?personaje2 - personaje_guardia1 ?arm - arm_pesada ?place ?place2 - place		
Precondiciones	<p>En el estado debe haber dos personajes de la Guardia Real vivos, uno no principal y otro principal. El personaje no principal tiene que tener un arma. El personaje principal debe tener la necesidad de tener esa arma que tiene su compañero. Además, los personajes deben estar en sitios distintos.</p> <p>El personaje que tiene el arma o no tiene que disparar o sí tiene que disparar, pero en el lugar donde se encuentra no quedan personajes de la Senda Dorada.</p>		
Efectos	<p>El efecto de esta acción es que el personaje que tiene el arma deja de estar en su lugar y se mueve al lugar del personaje que quiere el arma.</p> <p>Si el personaje tenía que disparar, (y se ha movido porque no quedan enemigos) deja de tener el deber de disparar.</p>		

Tabla 70: ACC-028

Nombre	ir-a-5-1	Código	ACC-029
Descripción	Mover a un personaje de la Senda Dorada, que quiere un arma, hacia el lugar donde se encuentra otro personaje que tiene el arma.		
Parámetros	?personaje - personaje_senda ?personaje2 - personaje ?arm - arm_pesada ?place ?place2 - place ?c1 ?c2 ?c3 ?c4 - contador		
Precondiciones	Para que esta acción se lleve a cabo, en el estado tiene que haber un personaje de la Senda Dorada vivo. Este personaje debe desear tener un arma que tenga otro personaje. Los personajes deben estar en sitios distintos.		
Efectos	El efecto de esta acción es que el personaje que quiere el arma deja de estar en su lugar y se mueve al lugar del personaje que tiene el arma. Además, en el lugar de salida habrá un personaje menos de la Senda Dorada y en el lugar de destino habrá un personaje más.		

Tabla 71: ACC-029

Nombre	ir-a-5-2	Código	ACC-030
Descripción	Mover a un personaje no principal de la Guardia Real que quiere un arma hacia el lugar donde se encuentra otro personaje que tiene el arma.		
Parámetros	?personaje - personaje_guardia2 ?personaje2 - personaje ?arm - arm_pesada ?place ?place2 - place		
Precondiciones	<p>Para que esta acción se lleve a cabo, en el estado tiene que haber, un personaje no principal de la Guardia Real vivo. Este personaje debe desear tener un arma que tenga otro personaje.</p> <p>Los personajes deben estar en sitios distintos.</p> <p>El personaje que desea tener el arma o no tiene que disparar o sí tiene que disparar, pero en el lugar donde se encuentra no quedan personajes de la Senda Dorada.</p>		
Efectos	<p>El efecto de esta acción es que el personaje que quiere el arma deja de estar en su lugar y se mueve al lugar del personaje que tiene el arma.</p> <p>Si el personaje tenía que disparar, (y se ha movido porque no quedan enemigos) deja de tener el deber de disparar.</p>		

Tabla 72: ACC-030

Nombre	ir-a-6	Código	ACC-031
Descripción	Un personaje de la Guardia Real se mueve hacia el lugar donde se encuentra un personaje de la Senda Dorada		
Parámetros	?personaje1 - personaje_guardia2 ?personaje2 - personaje_senda ?place ?place2 - place		
Precondiciones	Hay un personaje no principal de la Guardia Real vivo. Además, hay un personaje de la Senda Dorada vivo. Los personajes están en lugares distintos. El personaje de la Guardia real o no tiene que disparar o sí tiene que disparar, pero en el lugar donde se encuentra no quedan personajes de la Senda Dorada.		
Efectos	Como efectos se tiene: Si el personaje tenía que disparar, (y se ha movido porque no quedan enemigos) deja de tener el deber de disparar. El personaje de la Guardia Real abandona su lugar y se mueve hasta el lugar del personaje de la Senda Dorada. Por último, el personaje de la Guardia Real tendrá el deber de disparar.		

Tabla 73: ACC-031

Nombre	ir-a-7	Código	ACC-032
Descripción	Un personaje de la Senda Dorada, que tiene como misión matar a un personaje principal de la facción contraria, se mueve al lugar de ese personaje porque tiene el arma necesaria para acabar con dicho personaje.		
Parámetros	?personaje1 - personaje_senda ?personaje2 - personaje_guardia1 ?place1 – place ?place2 - place ?arm - arm_pesada ?ms3 - mision_SendaDorada3 ?c1 ?c2 ?c3 ?c4 - contador		
Precondiciones	<p>Para ejecutar esta acción, debe haber un personaje de la Senda Dorada vivo y un personaje principal de la Guardia Real. Además, estos personajes deben estar en lugares distintos.</p> <p>El personaje de la Senda Dorada tiene que tener una misión del tipo “mision_SendaDorada3”, la cual en su descripción indique que se ha de matar al personaje principal de la Guardia Real. Se debe cumplir que el personaje de la Senda Dorada tiene el arma con el que el personaje principal de la Guardia Real puede ser abatido.</p>		
Efectos	<p>Como efecto de la acción, se obtiene que el personaje que tiene la misión de matar al otro personaje, se mueve al lugar del personaje.</p> <p>Además, en el lugar de salida habrá un personaje menos de la Senda Dorada y en el lugar de destino habrá un personaje más.</p>		

Tabla 74: ACC-032

5.5.6. Ataques entre personajes

Nombre	ataqueFusil-1	Código	ACC-033
Descripción	Un personaje de la Guardia Real dispara a un personaje de la Senda Dorada, restándole vida.		
Parámetros	?personaje1 - personaje_guardia ?personaje2 - personaje_senda ?place - place ?arm - arm_ligera_guardia ?c1 ?c2 ?c3 - contador		
Precondiciones	Para ejecutar esta acción debe haber en el estado un personaje de la Senda Dorada vivo y un personaje de la Guardia Real vivo en el mismo lugar. El personaje de la Guardia Real tiene que tener un arma ligera. El personaje de la Senda Dorada tiene que tener un contador de vida quitada. Este contador no puede ser el inmediatamente anterior a la vida total del personaje.		
Efectos	El efecto es que al personaje de la Senda Real se le actualiza el valor del contador de la vida que lleva quitada. Se comprueba si el personaje de la Guardia Real que dispara se había desplazado al lugar para disparar, en cuyo caso se le quita el deber de disparar.		

Tabla 75: ACC-033

Nombre	ataqueFusil-2	Código	ACC-034
Descripción	Un personaje de la Senda Dorada dispara a un personaje de la Guardia Real no principal, restándole vida.		
Parámetros	?personaje1 - personaje_senda ?personaje2 - personaje_guardia2 ?place - place ?arm - arm_ligera_senda ?c1 ?c2 ?c3 - contador		
Precondiciones	<p>Para ejecutar esta acción debe haber en el estado un personaje de la Senda Dorada vivo y un personaje de la Guardia Real no principal vivo en el mismo lugar.</p> <p>El personaje de la Senda Dorada tiene que tener un arma ligera.</p> <p>El personaje de la Guardia real tiene que tener un contador de vida quitada. Este contador no puede ser el inmediatamente anterior a la vida total del personaje.</p>		
Efectos	El efecto es que al personaje de la Guardia Real se le actualiza el valor del contador de la vida que lleva quitada.		

Tabla 76: ACC-034

Nombre	Matar1	Código	ACC-035
Descripción	Un personaje de la Guardia Real mata a un personaje de la Senda Dorada que está haciendo una misión.		
Parámetros	?personaje1 - personaje_guardia ?personaje2 - personaje_senda ?place - place ?arm - arm_ligera_guardia ?m - mision_SendaDorada ?c1 ?c2 ?c3 ?c4 - contador		
Precondiciones	<p>Debe haber un personaje de la Guardia Real vivo y con un arma en el mismo sitio que un personaje de la Senda Dorada. El personaje de la Senda debe estar ocupado, teniendo el deber de realizar una misión.</p> <p>El personaje de la Senda Dorada tiene que tener un contador de vida quitada y un contador de vida total del personaje. El contador de vida quitada debe ser justo una unidad menor que el contador que lleva la vida total del personaje.</p>		
Efectos	<p>Como efecto se tiene que el personaje de la Senda Dorada acaba muerto. Esto hace que la misión que tenía que hacer quede desbloqueada.</p> <p>El número de personajes de la Senda Dorada que hay en el lugar se reduce una unidad.</p> <p>Se comprueba si el personaje de la Guardia Real que dispara se había desplazado al lugar para disparar, en cuyo caso se le quita el deber de disparar.</p>		

Tabla 77: ACC-035

Nombre	Matar2	Código	ACC-036
Descripción	Un personaje de la Guardia Real mata a un personaje de la Senda Dorada que no está haciendo ninguna misión.		
Parámetros	<p>?personaje1 - personaje_guardia</p> <p>?personaje2 - personaje_senda</p> <p>?place - place</p> <p>?arm - arm_ligera_guardia</p> <p>?c1 ?c2 ?C3 ?C4 - contador</p>		
Precondiciones	<p>Debe haber un personaje de la Guardia Real vivo y con un arma, en el mismo sitio que un personaje de la Senda Dorada. El personaje de la Senda debe no estar ocupado.</p> <p>El personaje de la Senda Dorada tiene que tener un contador de vida quitada y un contador de vida total del personaje. El contador de vida quitada debe ser justo una unidad menor que el contador que lleva la vida total del personaje.</p>		
Efectos	<p>Como efecto se tiene que el personaje de la Senda Dorada acaba muerto.</p> <p>El número de personajes de la Senda Dorada que hay en el lugar se reduce una unidad.</p> <p>Se comprueba si el personaje de la Guardia Real que dispara se había desplazado al lugar para disparar, en cuyo caso se le quita el deber de disparar.</p>		

Tabla 78: ACC-036

Nombre	Matar3	Código	ACC-037
Descripción	Un personaje de la Senda mata a un personaje no principal de la Guardia.		
Parámetros	?personaje1 - personaje_senda ?personaje2 - personaje_guardia2 ?place - place ?arm - arm_ligera_senda ?c1 ?c2 - contador		
Precondiciones	<p>Debe haber un personaje de la Senda Dorada vivo y con un arma, en el mismo sitio que un personaje no principal de la Guardia Real.</p> <p>El personaje de la Guardia real tiene que tener un contador de vida quitada y un contador de vida total del personaje. El contador de vida quitada, debe ser justo una unidad menor que el contador que lleva la vida total del personaje.</p>		
Efectos	El efecto de la acción es que el personaje de la Guardia Real muere.		

Tabla 79: ACC-037

Nombre	Matar-principal	Código	ACC-038
Descripción	Un personaje de la Senda Dorada mata a un personaje principal de la Guardia Real. Por lo tanto, completa una misión del tipo matar a un personaje de la Guardia Real principal.		
Parámetros	?personaje1 - personaje_senda ?personaje2 - personaje_guardia1 ?place - place_guardia ?arm - arm_pesada ?ms3 - mision_SendaDorada3		
Precondiciones	<p>Para que se de esta acción en el estado tiene que haber un personaje vivo de la Senda Dorada. El personaje de la Senda Dorada tiene que tener una misión del tipo “mision_SendaDorada3”.</p> <p>También tiene que haber un personaje de la Guardia Real principal vivo.</p> <p>Los dos personajes tienen que estar en el mismo lugar.</p> <p>Por último, se tiene que dar que el personaje de la Senda tenga en su poder el arma pesada necesaria para acabar con el personaje de la Guardia Real.</p>		
Efectos	Como efecto de esta acción, el personaje de la Senda Dorada completa la misión del tipo “mision_SendaDorada3” y deja de tenerla. Además, este personaje deja de estar ocupado. Además, el personaje de la Guardia Real acaba muerto.		

Tabla 80: ACC-038

5.6. Creación de problemas

Los problemas que se van a utilizar en la experimentación para poner a prueba el dominio implementado van a tener todos unos estados iniciales comunes. Un problema se diferenciará de otro en los objetivos teniendo cada uno una meta distinta.

A continuación, se hace una descripción de los diferentes objetos y predicados iniciales que tendrán los problemas.

La primera parte de los problemas a la que se hace referencia es a la que tiene que ver con la declaración de las instancias de los objetos.

(:objects

;LUGARES

Banapur - place_senda

Varshakot Tirtha Shanath - place_const

Fortaleza_Yuma Fortaleza_Paul Fortaleza_Pagan - place_guardia

;PERSONAJES

Sabal Ajay Amita - personaje_senda

Pagan Yuma Paul - personaje_guardia1

TenienteYuma TenientePaul TenientePagan - personaje_guardia2

;CONSTRUCCIONES

Torre Fabrica Iglesia – construcción

;ARMAS

AK-47 MP44 MP5 - arm_ligera_senda

M16 M14 RPG - arm_ligera_guardia

AK-74 MiniUzi G36 - arm_ligera_guardia

c4 - arm_explosiva

Antidoto Explosivo Armamento - arm_pesada

;MISIONES

LiberarKirat - mision_SendaDorada1

*DestruirLaTorre DestruirLaFabrica LiberarLaIglesia -
mision_SendaDorada2*

MatarAPagan MatarAYuma MatarAPaul - mision_SendaDorada3

ConquistarKirat - mision_GuardiaReal1

SalvarAYuma SalvarAPaul SalvarAPagan - mision_GuardiaReal2

)

Cabe destacar que en el dominio se especifican algunas instancias que ya no hay que declararlas en los problemas.

```
(:constants
    zero one two three - contador
    Destruida Liberada - estado_final
    politica religiosa - corriente
)
```

Una vez quedan definidas las distintas instancias que se pueden usar para cada objeto, se procede a comentar los predicados iniciales que componen el estado inicial de los problemas.

Primero especificamos el orden de los contadores.

```
(next1 zero one)
(next1 one two)
(next1 two three)
```

Luego inicializamos el número de personajes de la Senda Dorada que habrá en cada lugar, siendo “zero” en todos los lugares menos en Banapur, refugio de la Senda Dorada al principio de las historias.

```
(numero-personajes-lugar-S Banapur three)
(numero-personajes-lugar-S Varshakot zero)
(numero-personajes-lugar-S Tirtha zero)
(numero-personajes-lugar-S Shanath zero)
(numero-personajes-lugar-S Fortaleza_Pagan zero)
(numero-personajes-lugar-S Fortaleza_Paul zero)
(numero-personajes-lugar-S Fortaleza_Yuma zero)
```

A continuación, analizamos los que tienen que ver con las misiones de la Senda Dorada.

```
(mision-corriente DestruirLaTorre politica)
(mision-corriente DestruirLaFabrica politica)
(mision-corriente LiberarLaIglesia religiosa)

(misionS2-descripcion DestruirLaTorre Torre Destruida)
(misionS2-descripcion DestruirLaFabrica Fabrica Destruida)
(misionS2-descripcion LiberarLaIglesia Iglesia Liberada)
(misionS3-descripcion MatarAYuma Yuma)
(misionS3-descripcion MatarAPaul Paul)
(misionS3-descripcion MatarAPagan Pagan)

(mision-relaccionada LiberarKirat MatarAYuma)
(mision-relaccionada LiberarKirat MatarAPaul)
(mision-relaccionada LiberarKirat MatarAPagan)
(mision-relaccionada LiberarKirat DestruirLaTorre)
(mision-relaccionada LiberarKirat DestruirLaFabrica)
(mision-relaccionada LiberarKirat LiberarLaIglesia)
```

Lo primero a cada misión del tipo destruir o liberar una construcción se le asigna una corriente. Luego declaramos las distintas descripciones de las misiones. Primero las que tienen que ver con liberar y destruir construcciones y luego las que tienen que ver con matar personajes. Después declaramos que la misión de liberar Kirat está relacionada con todas las misiones de la Senda Dorada que no son del mismo tipo que ella.

Ahora los predicados iniciales de las misiones de la guardia real.

(misionG2-descripcion SalvarAYuma Yuma)
(misionG2-descripcion SalvarAPaul Paul)
(misionG2-descripcion SalvarAPagan Pagan)

En estas misiones solo es necesario indicar la descripción de cada una de ellas. Además, únicamente habrá misiones de salvar a personajes principales.

Luego especificamos las distintas construcciones que habrá en la región de Kirat, situándolas en distintos lugares y asignándolas un arma pesada a cada una. A las construcciones también les especificaremos el número de soldados que están defendiéndola y el número de soldados muertos que están defendiéndola.

;TORRE
(at Torre Tirtha)
(have-an-obj Torre Antidoto)
(construccion-soldados Torre two)
(construccion-soldados-muertos Torre zero)

;FABRICA
(at Fabrica Shanath)
(have-an-obj Fabrica Armamento)
(construccion-soldados Fabrica two)
(construccion-soldados-muertos Fabrica zero)

;IGLESIA
(at Iglesia Varshakot)
(have-an-obj Iglesia Explosivo)
(construccion-soldados Iglesia two)
(construccion-soldados-muertos Iglesia zero)

Ahora analizaremos los predicados iniciales de cada uno de los personajes que nos encontramos en el problema.

Personaje Ajay

(at Ajay Banapur)
(personaje-corriente Ajay politica)
(personaje-corriente Ajay religiosa)
(have-an-obj Ajay AK-47)
(have-an-obj Ajay c4)
(personaje-vida Ajay two)
(personaje-vida-quitada Ajay zero)

Personaje Sabal

(at Sabal Banapur)
(have-mision Sabal LiberarKirat)
(personaje-corriente Sabal religiosa)
(have-an-obj Sabal MP44)
(personaje-vida Sabal two)
(personaje-vida-quitada Sabal zero)

Personaje Amita

(at Amita Banapur)
(have-mision Amita LiberarKirat)
(personaje-corriente Amita politica)
(have-an-obj Amita MP5)
(personaje-vida Amita two)
(personaje-vida-quitada Amita zero)

Ajay, Sabal y Amita estarán situados inicialmente en Banapur. Sabal y Amita tendrán la mision LiberarKirat que está relacionada con las demás misiones de la Senda Dorada. Además, Sabal y Amita seguirán la corriente de la religión y la política respectivamente. Por su parte Ajay no tendrá ninguna misión, pero seguirá tanto la corriente política y religiosa de la Senda Dorada. Los tres personajes tendrán un fusil, y su vida inicial vendrá especificada por el contador two, y como es el estado inicial no le habrán quitado vida a ninguno.

Personaje Pagan

(personaje-principal Pagan)
(at Pagan Fortaleza_Pagan)
(criptonita Pagan Armamento)
(have-an-obj Pagan M16)
(have-mision Pagan SalvarAPagan)

Personaje Yuma

(at Yuma Fortaleza_Yuma)
(criptonita Yuma Antidoto)
(have-an-obj Yuma M14)
(have-mision Yuma SalvarAYuma)

Personaje Paul

(personaje-principal Paul)
(at Paul Fortaleza_Paul)
(criptonita Paul Explosivo)
(have-an-obj Paul RPG)
(have-mision Paul SalvarAPaul)

Yuma Paul y Pagan estarán situados en la Fortaleza de Yuma, la Fortaleza de Paul y la Fortaleza de Pagan respectivamente.

Los tres inmóviles en su fortaleza están a salvo de cualquier peligro, pero estas fortalezas no son del todo seguras. Existe para cada una un arma pesada con la que poder entrar y acabar con el personaje que se esconde en ellas. Para acabar con Yuma habrá que conseguir un Antídoto, para acabar con Paul un Explosivo y por último para acabar con Pagan es necesario un fuerte Armamento. Estos personajes querrán salvarse consiguiendo las armas mencionadas, además irán armados con distintos fusiles

TenientePaul

(at TenientePaul Fortaleza_Paul)
(have-an-obj TenientePaul AK-74)
(personaje-vida TenientePaul two)
(personaje-vida-quitada TenientePaul zero)
(have-mision TenientePaul SalvarAPaul)

TenientePagan

(at TenientePagan Fortaleza_Pagan)
(have-an-obj TenientePagan MiniUzi)
(personaje-vida TenientePagan two)
(personaje-vida-quitada TenientePagan zero)
(have-mision TenientePagan SalvarAPagan)

TenienteYuma

(at TenienteYuma Fortaleza_Yuma)
(have-an-obj TenienteYuma G36)
(personaje-vida TenienteYuma two)
(personaje-vida-quitada TenienteYuma zero)
(have-mision TenienteYuma SalvarAYuma)

TenientePaul, TenientePagan y TenienteYuma, estarán situados en la Fortaleza de Yuma, la Fortaleza de Paul y la Fortaleza de Pagan respectivamente. Los tres tendrán la misión de salvar a sus respectivos jefes e irán armados con un tipo distinto de fusil cada uno.

5.7. Creación de frases

A las acciones implementadas en el dominio se las puede cargar de significado narrativo. Para conseguirlo vamos a utilizar la función *narrate* del planificador Sayphi, con la cual podemos convertir una acción en una oración. Para ello utilizamos el nombre y los parámetros de la acción.

Con esto conseguimos que, al crear un plan, utilizando el dominio y un problema, se generen una secuencia de oraciones, fáciles de leer y entender.

A continuación, se muestra cómo quedará la transcripción de cada una de las acciones implementadas en nuestro dominio:

5.7.1. Misiones

- **mandar-mision1:** SABAL le pide a AJAY que le ayude a LIBERARLAIGLESIA de Kirat, AJAY decidido a ayudar a su compañero a recuperar la cultura RELIGIOSA de Kirat acepta la misión.
- **mandar-mision2:** SABAL consulta a AJAY si sería capaz de MATARAPAGAN, uno de los líderes que tiene oprimido al pueblo de Kirat, AJAY sin dudarlo se embarca en la misión
- **aprender-mision1:** SABAL desea recuperar la cultura RELIGIOSA de Kirat, por lo que va a intentar LIBERARLAIGLESIA de Kirat, que controla la Guardia Real
- **aprender-mision2:** SABAL cansado de la opresión bajo la que se encuentran los ciudadanos de Kirat, va a intentar MATARAPPAUL acabando así con uno de los líderes de la dictadura.
- **aprender-misionSenda-principal:** Muertos los líderes de la Senda Dorada AMITA y SABAL. AJAY decide vengarles, intentando LIBERARKIRAT por solitario

5.7.2. Corrientes

- **aprender-corriente1:** SABAL triste por la muerte de su camarada AMITA intentara cumplir su deseo de recuperar la cultura Política de Kirat
- **aprender-corriente2:** AMITA triste por la muerte de su camarada SABAL intentara cumplir su deseo de recuperar la cultura Religiosa de Kirat

5.7.3. Movimientos

- **ir-a-1:** AMITA se dirige de TIRTHA a la FABRICA de SHANATH para llevar a cabo su misión DESTRUIRLAFABRICA
- **ir-a-2-1:** AMITA se dirige de BANAPUR a la FABRICA de SHANATH dado que allí se sitúa el eficaz ARMAMENTO que necesita
- **ir-a-2-2:** TENIENTEPAUL se dirige de BANAPUR a la IGLESIA de VARSHAKOT dado que allí se sitúa el eficaz EXPLOSIVO que necesita
- **ir-a-3:** AJAY se dirige de BANAPUR a TIRTHA porque quiere darle ANTIDOTO a su compañero SABAL
- **ir-a-4-1:** TENIENTEPAUL se dirige de BANAPUR a FORTALEZA_YUMA porque quiere darle ANTIDOTO a su compañero YUMA
- **ir-a-4-2:** TENIENTEYUMA se dirige de TIRTHA a FORTALEZA_YUMA porque quiere darle ANTIDOTO a su líder YUMA

- **ir-a-5-1:** TENIENTEPAUL se dirige de BANAPUR a VARSHAKOT porque quiere ANTIDOTO que lo tiene SABAL
- **ir-a-5-2:** TENIENTEPAUL se dirige de BANAPUR a VARSHAKOT porque quiere ANTIDOTO que lo tiene SABAL
- **ir-a-6:** TENIENTEYUMA se dirige de FORTALEZA_YUMA a VARSHAKOT para atacar a un miembro de la Senda Dorada
- **ir-a-7:** AMITA se dirige de TIRTHA a FORTALEZA_YUMA con el letal ANTIDOTO para acabar con YUMA

5.7.4. Conseguir armas

- **querer-objeto-por-mision-senda:** SABAL sabe que si quiere MATARAPUL necesitara tener en su poder un fuerte EXPLOSIVO, por lo que quiere conseguirlo antes de ir a por PAUL
- **querer-objeto-por-mision-guardia1:** El ruin YUMA quiere el potente ANTIDOTO para salvar su vida
- **querer-objeto-por-mision-guardia2:** El miserable TENIENTEYUMA quiere conseguir el ANTIDOTO que podría acabar con su líder YUMA
- **coger-objeto-Senda:** En la TORRE de TIRTHA. AMITA coge un mortífero ANTIDOTO
- **coger-objeto-Guardia:** En la IGLESIA de VARSHAKOT, TENIENTEYUMA coge un formidable EXPLOSIVO
- **dar-objeto-Senda:** En SHANATH, AJAY le da ANTIDOTO a SABAL, el cual necesitaba para cumplir con su objetivo
- **dar-objeto-Guardia1:** En FORTALEZA_PAUL, TENIENTEPAUL le da EXPLOSIVO a su líder PAUL
- **dar-objeto-Guardia2:** En SHANATH, TENIENTEPAUL le da ANTIDOTO a su socio TENIENTEYUMA
- **coger-objeto-de-personaje-muerto:** En SHANATH, AMITA coge ARMAMENTO del cuerpo sin vida de TENIENTEPAUL
- **destruir-criptonita:** YUMA consigue salvar su vida destruyendo el abrumador ANTIDOTO. Destruído el artefacto, la Senda Dorada ya no podrá acabar con él

5.7.5. Ataques a construcciones

- **ataqueSendaArmaCortaEdificio:** En la IGLESIA de VARSHAKOT, AJAY utiliza su AK-47 para acabar con un soldado
- **ataqueSendaExplosivosEdificio:** En la TORRE de TIRTHA, AJAY utiliza C4 para acabar con todos los soldados
- **destruir-construccion1:** En VARSHAKOT, SABAL consigue que la IGLESIA quede LIBERADA completando así la misión de LIBERARLAIGLESIA
- **destruir-construccion2:** En VARSHAKOT, AJAY consigue que la IGLESIA quede LIBERADA completando así su misión LIBERARLAIGLESIA
- **destruir-construccion3:** En FABRICA, SABAL consigue que la FABRICA quede DESTRUIDA completando así la misión DESTRUIRLAFABRICA que quería realizar su compañero

5.7.6. Ataques entre personajes

- **ataqueFusil-1:** *TENIENTEYUMA dispara AJAY con su G36 en BANAPUR*
- **ataqueFusil-2:** *AMITA dispara TENIENTEPAUL con su MP5 en BANAPUR*
- **matar1:** *TENIENTEPAUL dispara AJAY con su AK-74, AJAY no aguanta más disparos y muere en TIRTHA*
- **matar2:** *TENIENTEPAUL dispara AJAY con su AK-74, AJAY no aguanta más disparos y muere en TIRTHA*
- **matar3:** *AJAY dispara TENIENTEYUMA con su AK-47, TENIENTEYUMA no aguanta más disparos y muere en BANAPUR*
- **matar-principal:** *En FORTALEZA_PAGAN, AJAY mata PAGAN utilizando el gran ARMAMENTO*

6. Experimentación

Una de las partes más importantes de este trabajo consiste en analizar el funcionamiento del planificador con el dominio implementado.

Este apartado se divide en tres subapartados. En el primer subapartado se define el plan de pruebas que se va a realizar. En el siguiente subapartado se describen los distintos problemas con los que se realiza la experimentación y se muestran los resultados obtenidos de cada problema. Por último, en el tercer subapartado se analizan los resultados obtenidos.

6.1. Plan de pruebas

Lo que vamos a analizar en esta experimentación es la variabilidad que presentan los planes generados por el planificador para un problema, utilizando el dominio implementado.

También analizaremos la coherencia que presenta cada plan y analizaremos si influye a la coherencia de las historias la probabilidad de que el planificador pueda elegir acciones de forma aleatoria. Se dice que un plan es coherente si las acciones que se realizan en el plan se realizan cuando realmente pueden realizarse. Por ejemplo, si en nuestros planes aparecieran acciones de ataques entre personajes que no están en el mismo sitio, entonces estas historias no serían coherentes porque en la definición de las acciones se ha definido que para que se realice un ataque entre personajes ambos personajes deben estar en el mismo sitio.

Para llevar a cabo el análisis, utilizaremos una serie de problemas, con un estado inicial común, por lo que se diferenciarán unos de otros en los objetivos, teniendo cada uno el objetivo de llegar a un estado final concreto. Además, el planificador se va a ejecutar utilizando configuraciones de aleatoriedad distintas y se realizará un estudio de cuál es el efecto de aumentar la probabilidad de aleatoriedad en la planificación.

Los pasos que se llevarán a cabo para analizar los planes que se generan serán, para cada problema:

- Ejecutar el problema cuatro veces con una probabilidad de aleatoriedad baja, del 0.001%.
 - Almacenar la longitud de cada plan y la diversidad que presenta cada plan con respecto al plan generado anteriormente.
- Ejecutar el problema cuatro veces con una probabilidad de aleatoriedad baja, del 20%.
 - Almacenar la longitud de cada plan y la diversidad que presenta cada plan con respecto al plan generado anteriormente.
- Ejecutar el problema cuatro veces con una probabilidad de aleatoriedad media, del 50%.
 - Almacenar la longitud de cada plan y la diversidad que presenta cada plan con respecto al plan generado anteriormente.

- Ejecutar el problema cuatro veces con una probabilidad de aleatoriedad alta, del 60%.
 - Almacenar la longitud de cada plan y la diversidad que presenta cada plan con respecto al plan generado anteriormente.

La probabilidad de aleatoriedad la utilizaremos gracias al parámetro `probability-ehc` (que toma valores entre 0 y 1), del algoritmo `rrt`, que incluye el planificador `Sayphi`.

Por su parte la longitud y diversidad de cada plan nos ayudarán a analizar cuál es la variabilidad que presentan los planes que genera nuestro dominio. Estos dos rasgos serán analizados entre planes obtenidos de un mismo problema y con una probabilidad de aleatoriedad igual. La longitud es el número de acciones totales que aparecen en un plan. La diversidad que muestra un plan A frente a un plan B se calcula como la suma del número de acciones que aparecen en A y no en B y las que aparecen en B y no en A.

Para analizar la coherencia de los planes, leeremos las historias que se generan y compararemos los resultados obtenidos con lo que se esperaba obtener, y si todo marcha según la definición e intención de cada acción, se podrá concluir que para el dominio implementado se generan planes o historias coherentes.

6.2. Problemas

A continuación, se presentan los resultados obtenidos en la experimentación. Para llevar a cabo la experimentación se han creado 10 problemas de diferente dificultad. Para cada problema se va a describir su definición (objetivos que se tiene que alcanzar en las historias) y se van a mostrar sus resultados. Dentro de los resultados se va a mostrar uno o varios ejemplos de las historias que genera el planificador y una tabla con los datos que se obtienen tras la ejecución del problema tal y como se ha definido en el plan de pruebas.

Como ya se ha indicado para cada problema mostraremos un ejemplo de las historias que genera el planificador. Para hacer más fácil la lectura de estas historias se han marcado en verde las acciones que hacen que los objetivos del problema se completen.

Los datos definidos en el plan de pruebas van a ser recogidos para cada problema en una tabla como la que se muestra a continuación.

aleatoriedad (0.001%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan					
Longitud					
Diversidad					
aleatoriedad (20%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan					
Longitud					
Diversidad					
aleatoriedad (0.50%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan					
Longitud					
Diversidad					
aleatoriedad (0.70%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan					
Longitud					
Diversidad					

Tabla 81: Ejemplo Tabla de Datos para Problemas

Esta tabla se divide en cuatro tablas. En cada tabla los datos que se van a almacenar estarán condicionados por la probabilidad de aleatoriedad con la que se ejecuta el planificador (0,001%, 20%, 50% y 70%).

6.2.1. Problema 1

Definición

La definición del problema 1 es:

```
(:goal (and
      (mision-complete LiberarLaIglesia)
      (mision-complete DestruirLaFabrica)
      (mision-complete DestruirLaTorre)
    )
)
```

Para completar el problema se deben completar las tres misiones “LiberarLaIglesia”, “DestruirLaFabrica” y “DestruirLaTorre”. Para que esto pase las misiones tienen que ser asignadas a los personajes de la Senda Dorada y éstos deben realizar las acciones oportunas para completarlas con éxito.

Resultado

A continuación, se muestra uno de los planes encontrados por el planificador para el problema 1. En concreto este problema se ha obtenido aplicando una probabilidad de aleatoriedad del 20%.

Plan 1

SABAL le pide a AJAY que le ayude a LIBERARLAIGLESIA de Kirat, AJAY decidido a ayudar a su compañero a recuperar la cultura RELIGIOSA de Kirat acepta la misión AJAY se dirige de BANAPUR a la IGLESIA de VARSHAKOT para llevar a cabo su misión LIBERARLAIGLESIA

TENIENTEYUMA se dirige de FORTALEZA_YUMA a VARSHAKOT para atacar a un miembro de la Senda Dorada

SABAL cansado de la opresión bajo la que se encuentran los ciudadanos de Kirat, va a intentar MATARAPAUl acabando así con uno de los líderes de la dictadura

El miserable TENIENTEYUMA quiere conseguir el ANTIDOTO que podría acabar con su líder YUMA

TENIENTEPAUL se dirige de FORTALEZA_PAUL a BANAPUR para atacar a un miembro de la Senda Dorada

El ruin PAUL quiere el potente EXPLOSIVO para salvar su vida

AMITA cansada de la opresión bajo la que se encuentran los ciudadanos de Kirat, va a intentar MATARAPAGAN acabando así con uno de los líderes de la dictadura

SABAL sabe que si quiere MATARAPAUl necesitara tener en su poder un fuerte EXPLOSIVO, por lo que quiere conseguirlo antes de ir a por PAUL

AMITA sabe que si quiere MATARAPAGAN necesitara tener en su poder un fuerte ARMAMENTO, por lo que quiere conseguirlo antes de ir a por PAGAN

En la IGLESIA de VARSHAKOT, AJAY utiliza su AK-47 para acabar con un soldado

En la IGLESIA de VARSHAKOT, TENIENTEYUMA coge un formidable EXPLOSIVO

AMITA dispara TENIENTEPAUL con su MP5 en BANAPUR

En la IGLESIA de VARSHAKOT, AJAY utiliza su AK-47 para acabar con un soldado

En VARSHAKOT, AJAY consigue que la IGLESIA quede LIBERADA completando así su misión LIBERARLAIGLESIA

AMITA le pide a AJAY que le ayude a DESTRUIRLATORRE de Kirat, AJAY decidido a ayudar a su compañero a recuperar la cultura POLITICA de Kirat acepta la misión

AJAY se dirige de VARSHAKOT a la TORRE de TIRTHA para llevar a cabo su misión DESTRUIRLATORRE

En la TORRE de TIRTHA, AJAY utiliza C4 para acabar con todos los soldados

AMITA se dirige de BANAPUR a la FABRICA de SHANATH dado que allí se sitúa el eficaz ARMAMENTO que necesita

En la FABRICA de SHANATH, AMITA utiliza su MP5 para acabar con un soldado

En la FABRICA de SHANATH, AMITA utiliza su MP5 para acabar con un soldado

En SHANATH, AMITA consigue que la FABRICA quede DESTRUIDA completando así la misión de DESTRUIRLAFABRICA

En TIRTHA, AJAY consigue que la TORRE quede DESTRUIDA completando así su misión DESTRUIRLATORRE

End.

Datos Obtenidos para el Problema 1

aleatoriedad (0.001%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	Si	Si	Si	Si	-
Longitud	15	15	15	15	15
Diversidad	-	0	0	0	0
aleatoriedad (20%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	Si	Si	Si	Si	-
Longitud	23	18	15	18	18
Diversidad	-	33	3	3	13
aleatoriedad (0.50%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	Si	Si	Si	Si	-
Longitud	24	26	23	28	25
Diversidad	-	34	25	27	29
aleatoriedad (0.70%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	No	Si	Si	Si	-
Longitud	41	35	45	30	37
Diversidad	-	60	62	67	64

Tabla 82: Datos Problema 1

6.2.1. Problema 2

Definición

La definición del problema 2 es:

```
(:goal (and
      ; Acaban muertos todos los malos;
      (mision-complete MatarAPagan)
      (mision-complete MatarAPaul)
      (mision-complete MatarAYuma)
    )
)
```

Para completar el problema se deben completar las tres misiones “MatarAPagan”, “MatarAPaul” y “MatarAYuma”. Para que esto pase las misiones tienen que ser asignadas a los personajes de la Senda Dorada y además estos deben realizar distintas acciones para conseguir las diferentes armas con las que poder completar las misiones.

Resultado

A continuación, se muestra uno de los planes encontrados por el planificador para el problema 2. En concreto este problema se ha obtenido aplicando una probabilidad de aleatoriedad del 20%.

Plan 1

SABAL consulta a AJAY si sería capaz de MATARAPAGAN, uno de los líderes que tiene oprimido al pueblo de Kirat, AJAY sin dudarlo se embarca en la misión

SABAL cansado de la opresión bajo la que se encuentran los ciudadanos de Kirat, va a intentar MATARAPPAUL acabando así con uno de los líderes de la dictadura

TENIENTEPAUL se dirige de FORTALEZA_PAUL a BANAPUR para atacar a un miembro de la Senda Dorada

AMITA desea recuperar la cultura POLITICA de Kirat, por lo que va a intentar DESTRUIRLATORRE de Kirat, que controla la Guardia Real

El miserable TENIENTEPAUL quiere conseguir el EXPLOSIVO que podría acabar con su líder PAUL

El miserable TENIENTEPAGAN quiere conseguir el ARMAMENTO que podría acabar con su líder PAGAN

AMITA se dirige de BANAPUR a la TORRE de TIRTHA para llevar a cabo su misión DESTRUIRLATORRE

El ruin YUMA quiere el potente ANTIDOTO para salvar su vida

En la TORRE de TIRTHA, AMITA utiliza su MP5 para acabar con un soldado

AJAY sabe que si quiere MATARAPAGAN necesitara tener en su poder un fuerte ARMAMENTO, por lo que quiere conseguirlo antes de ir a por PAGAN

En la TORRE de TIRTHA, AMITA utiliza su MP5 para acabar con un soldado

En TIRTHA, AMITA consigue que la TORRE quede DESTRUIDA completando así su misión DESTRUIRLATORRE

AMITA cansada de la opresión bajo la que se encuentran los ciudadanos de Kirat, va a intentar MATARAYUMA acabando así con uno de los líderes de la dictadura

En la TORRE de TIRTHA. AMITA coge un mortífero ANTIDOTO
 SABAL sabe que si quiere MATARAPAUl necesitara tener en su poder un fuerte
 EXPLOSIVO, por lo que quiere conseguirlo antes de ir a por PAUL
 TENIENTEPAGAN se dirige de FORTALEZA_PAGAN a BANAPUR para atacar a un
 miembro de la Senda Dorada
 AMITA se dirige de TIRTHA a FORTALEZA_YUMA con el letal ANTIDOTO para acabar
 con YUMA
 SABAL dispara TENIENTEPAUL con su MP44 en BANAPUR
 En FORTALEZA_YUMA, AMITA mata YUMA utilizando el gran ANTIDOTO
 SABAL se dirige de BANAPUR a la IGLESIA de VARSHAKOT dado que allí se sitúa el eficaz
 EXPLOSIVO que necesita
 En la IGLESIA de VARSHAKOT, SABAL utiliza su MP44 para acabar con un soldado
 En la IGLESIA de VARSHAKOT, SABAL utiliza su MP44 para acabar con un soldado
 En VARSHAKOT, SABAL consigue que la IGLESIA quede LIBERADA completando así la
 misión de LIBERARLAIGLESIA
 En la IGLESIA de VARSHAKOT. SABAL coge un mortífero EXPLOSIVO
 AJAY se dirige de BANAPUR a la FABRICA de SHANATH dado que allí se sitúa el eficaz
 ARMAMENTO que necesita
 En la FABRICA de SHANATH, AJAY utiliza C4 para acabar con todos los soldados
 En SHANATH, AJAY consigue que la FABRICA quede DESTRUIDA completando así la
 misión de DESTRUIRLAFABRICA
 En la FABRICA de SHANATH. AJAY coge un mortífero ARMAMENTO
 SABAL se dirige de VARSHAKOT a FORTALEZA_PAUL con el letal EXPLOSIVO para
 acabar con PAUL
 AJAY se dirige de SHANATH a FORTALEZA_PAGAN con el letal ARMAMENTO para
 acabar con PAGAN
 En FORTALEZA_PAGAN, AJAY mata PAGAN utilizando el gran ARMAMENTO
 En FORTALEZA_PAUL, SABAL mata PAUL utilizando el gran EXPLOSIVO
 End.

Para el problema 2 también se han obtenido planes que no alcanzan los objetivos que
 se han definido. A continuación, se muestra un plan que no encontró solución para el
 problema 2. Para generar este plan se ejecutó el dominio con una probabilidad de
 aleatoriedad del 50%.

Plan 2

AMITA le pide a AJAY que le ayude a DESTRUIRLATORRE de Kirat, AJAY decidido a
 ayudar a su compañero a recuperar la cultura POLITICA de Kirat acepta la misión
 TENIENTEYUMA se dirige de FORTALEZA_YUMA a BANAPUR para atacar a un miembro
 de la Senda Dorada
 SABAL desea recuperar la cultura RELIGIOSA de Kirat, por lo que va a intentar
 LIBERARLAIGLESIA de Kirat, que controla la Guardia Real
 TENIENTEPAUL se dirige de FORTALEZA_PAUL a BANAPUR para atacar a un miembro
 de la Senda Dorada
 TENIENTEYUMA dispara AJAY con su G36 en BANAPUR

AMITA desea recuperar la cultura POLITICA de Kirat, por lo que va a intentar
 DESTRUIRLAFABRICA de Kirat, que controla la Guardia Real
 AMITA se dirige de BANAPUR a la FABRICA de SHANATH para llevar a cabo su misión
 DESTRUIRLAFABRICA
 SABAL dispara TENIENTEYUMA con su MP44 en BANAPUR
 El ruin PAGAN quiere el potente ARMAMENTO para salvar su vida
 El miserable TENIENTEPAUL quiere conseguir el EXPLOSIVO que podría acabar con su
 líder PAUL
 El ruin YUMA quiere el potente ANTIDOTO para salvar su vida
 TENIENTEPAGAN se dirige de FORTALEZA_PAGAN a BANAPUR para atacar a un
 miembro de la Senda Dorada
 El ruin PAUL quiere el potente EXPLOSIVO para salvar su vida
 TENIENTEPAUL dispara SABAL con su AK-74 en BANAPUR
 TENIENTEPAUL se dirige de BANAPUR a la IGLESIA de VARSHAKOT dado que allí se
 sitúa el eficaz EXPLOSIVO que necesita
 El miserable TENIENTEYUMA quiere conseguir el ANTIDOTO que podría acabar con su
 líder YUMA
 TENIENTEYUMA dispara SABAL con su G36, SABAL no aguanta más disparos y muere
 en BANAPUR
 En la IGLESIA de VARSHAKOT, TENIENTEPAUL coge un formidable EXPLOSIVO
 TENIENTEPAUL se dirige de VARSHAKOT a BANAPUR para atacar a un miembro de la
 Senda Dorada
 AJAY dispara TENIENTEPAUL con su AK-47 en BANAPUR
 AJAY dispara TENIENTEYUMA con su AK-47, TENIENTEYUMA no aguanta más disparos
 y muere en BANAPUR
 AJAY se dirige de BANAPUR a la TORRE de TIRTHA para llevar a cabo su misión
 DESTRUIRLATORRE
 AMITA triste por la muerte de su camarada SABAL intentara cumplir su deseo de
 recuperar la cultura Religiosa de Kirat
 TENIENTEPAUL se dirige de BANAPUR a TIRTHA para atacar a un miembro de la Senda
 Dorada En la TORRE de TIRTHA.
 TENIENTEPAUL coge un formidable ANTIDOTO
 En la TORRE de TIRTHA, AJAY utiliza C4 para acabar con todos los soldados
 TENIENTEPAUL dispara AJAY con su AK-74, AJAY no aguanta más disparos y muere en
 TIRTHA
 TENIENTEPAUL se dirige de TIRTHA a FORTALEZA_PAUL porque quiere darle EXPLOSIVO
 a su líder PAUL
 En la FABRICA de SHANATH, AMITA utiliza su MP5 para acabar con un soldado
 En FORTALEZA_PAUL, TENIENTEPAUL le da EXPLOSIVO a su líder PAUL
 ... to be continued.

Datos Obtenidos para el Problema 2

aleatoriedad (0,001%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	Si	Si	Si	Si	-
Longitud	32	32	32	32	32
Diversidad	-	0	0	0	0
aleatoriedad (20%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	Si	Si	Si	Si	-
Longitud	36	43	40	32	37
Diversidad	-	46	47	38	44
aleatoriedad (50%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	No	Si	No	No	-
Longitud	45	55	37	30	-
Diversidad	-	56	66	53	-
aleatoriedad (70%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	Si	No	Si	No	-
Longitud	53	36	44	40	48
Diversidad	-	73	48	64	-

Tabla 83: Datos Problema 2

6.2.3. Problema 3

Definición

La definición del problema 3 es:

```
(:goal (and
      ;+
      (mision-complete MatarAPagan)
      (mision-complete MatarAPaul)
      (mision-complete MatarAYuma)
      ;+
      (mision-complete LiberarLaIglesia)
      (mision-complete DestruirLaFabrica)
      (mision-complete DestruirLaTorre)
    )
)
```

Para completar el problema se deben completar las misiones que se han definido en el problema 1 y las que se han definido en el problema2 siendo este problema el conjunto de todas esas misiones.

Resultado

A continuación, se muestra uno de los planes encontrados por el planificador para el problema 3. En concreto este plan se ha obtenido aplicando una probabilidad de aleatoriedad del 0.001%.

Plan 1

SABAL le pide a AJAY que le ayude a LIBERARLAIGLESIA de Kirat, AJAY decidido a ayudar a su compañero a recuperar la cultura RELIGIOSA de Kirat acepta la misión AMITA desea recuperar la cultura POLITICA de Kirat, por lo que va a intentar DESTRUIRLATORRE de Kirat, que controla la Guardia Real

SABAL cansado de la opresión bajo la que se encuentran los ciudadanos de Kirat, va a intentar MATARAPAUl acabando así con uno de los líderes de la dictadura

El miserable TENIENTEYUMA quiere conseguir el ANTIDOTO que podría acabar con su líder YUMA

El miserable TENIENTEPAUL quiere conseguir el EXPLOSIVO que podría acabar con su líder PAUL

El miserable TENIENTEPAGAN quiere conseguir el ARMAMENTO que podría acabar con su líder PAGAN

AMITA se dirige de BANAPUR a la TORRE de TIRTHA para llevar a cabo su misión DESTRUIRLATORRE

En la TORRE de TIRTHA, AMITA utiliza su MP5 para acabar con un soldado

SABAL sabe que si quiere MATARAPAUl necesitara tener en su poder un fuerte EXPLOSIVO, por lo que quiere conseguirlo antes de ir a por PAUL

En la TORRE de TIRTHA, AMITA utiliza su MP5 para acabar con un soldado

En TIRTHA, AMITA consigue que la TORRE quede DESTRUIDA completando así su misión DESTRUIRLATORRE

En la TORRE de TIRTHA, AMITA coge un mortífero ANTIDOTO
 AJAY se dirige de BANAPUR a la IGLESIA de VARSHAKOT para llevar a cabo su misión
 LIBERARLAIGLESIA
 SABAL se dirige de BANAPUR a la IGLESIA de VARSHAKOT dado que allí se sitúa el eficaz
 EXPLOSIVO que necesita
 TENIENTEPAUL se dirige de FORTALEZA_PAUL a TIRTHA para atacar a un miembro de
 la Senda Dorada
 TENIENTEPAGAN se dirige de FORTALEZA_PAGAN a TIRTHA para atacar a un miembro
 de la Senda Dorada
 AMITA dispara TENIENTEPAGAN con su MP5 en TIRTHA
 AMITA cansado de la opresión bajo la que se encuentran los ciudadanos de Kirat, va a
 intentar MATARAYUMA acabando así con uno de los líderes de la dictadura
 En la IGLESIA de VARSHAKOT, AJAY utiliza C4 para acabar con todos los soldados
 En VARSHAKOT, AJAY consigue que la IGLESIA quede LIBERADA completando así su
 misión LIBERARLAIGLESIA
 AMITA le pide a AJAY que le ayude a DESTRUIRLAFABRICA de Kirat, AJAY decidido a
 ayudar a su compañero a recuperar la cultura POLITICA de Kirat acepta la misión
 AJAY se dirige de VARSHAKOT a la FABRICA de SHANATH para llevar a cabo su misión
 DESTRUIRLAFABRICA
 En la IGLESIA de VARSHAKOT, SABAL coge un mortífero EXPLOSIVO
 En la FABRICA de SHANATH, AJAY utiliza C4 para acabar con todos los soldados
 En SHANATH, AJAY consigue que la FABRICA quede DESTRUIDA completando así su
 misión DESTRUIRLAFABRICA
 SABAL consulta a AJAY si sería capaz de MATARAPAGAN, uno de los líderes que tiene
 oprimido al pueblo de Kirat, AJAY sin dudarlo se embarca en la misión
 En la FABRICA de SHANATH, AJAY coge un mortífero ARMAMENTO
 SABAL se dirige de VARSHAKOT a FORTALEZA_PAUL con el letal EXPLOSIVO para
 acabar con PAUL
 AJAY se dirige de SHANATH a FORTALEZA_PAGAN con el letal ARMAMENTO para
 acabar con PAGAN
 AMITA se dirige de TIRTHA a FORTALEZA_YUMA con el letal ANTIDOTO para acabar
 con YUMA
 En FORTALEZA_PAUL, SABAL mata PAUL utilizando el gran EXPLOSIVO
 En FORTALEZA_PAGAN, AJAY mata PAGAN utilizando el gran ARMAMENTO
 En FORTALEZA_YUMA, AMITA mata YUMA utilizando el gran ANTIDOTO
 End.

Datos Obtenidos para el Problema 3

aleatoriedad (0,001%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	Si	Si	Si	Si	-
Longitud	33	33	33	33	33
Diversidad	-	0	0	0	0
aleatoriedad (20%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	Si	Si	No	Si	-
Longitud	47	34	47	48	43
Diversidad	-	67	61	69	-
aleatoriedad (50%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	No	Si	No	No	-
Longitud	38	60	38	54	-
Diversidad	-	74	80	76	-
aleatoriedad (70%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	Si	No	No	Si	-
Longitud	61	54	46	46	53
Diversidad	-	97	72	52	-

Tabla 84: Datos Problema 3

6.2.4. Problema 4

Definición

La definición del problema 4 es:

```
(:goal (and
      ;Acaban muertos
      (personaje-muerto TenientePagan)
      (personaje-muerto TenienteYuma)
      (personaje-muerto Sabal)
      ;+
      (mision-complete MatarAPagan)
      (mision-complete MatarAPaul)
      (mision-complete MatarAYuma)
      ;+
      (mision-complete LiberarLaIglesia)
      (mision-complete DestruirLaFabrica)
      (mision-complete DestruirLaTorre)
    )
)
```

Este problema es el mismo que el problema 3 pero además se añade que los personajes TenientePagan, TenienteYuma y Sabal deben acabar muertos.

Resultado

A continuación, se muestra uno de los planes encontrados por el planificador para el problema 4. En concreto este problema se ha obtenido aplicando una probabilidad de aleatoriedad del 20%.

Plan 1

SABAL le pide a AJAY que le ayude a LIBERARLAIGLESIA de Kirat, AJAY decidido a ayudar a su compañero a recuperar la cultura RELIGIOSA de kirat acepta la misión AMITA desea recuperar la cultura POLITICA de Kirat, por lo que va a intentar DESTRUIRLATORRE de Kirat, que controla la Guardia Real

SABAL cansado de la opresión bajo la que se encuentran los ciudadanos de Kirat, va a intentar MATARAPAUl acabando así con uno de los líderes de la dictadura

TENIENTEPAUL se dirige de FORTALEZA_PAUL a BANAPUR para atacar a un miembro de la Senda Dorada

El miserable TENIENTEPAGAN quiere conseguir el ARMAMENTO que podría acabar con su líder PAGAN

AMITA se dirige de BANAPUR a la TORRE de TIRTHA para llevar a cabo su misión DESTRUIRLATORRE

TENIENTEYUMA se dirige de FORTALEZA_YUMA a BANAPUR para atacar a un miembro de la Senda Dorada

El miserable TENIENTEPAUL quiere conseguir el EXPLOSIVO que podría acabar con su líder PAUL

SABAL sabe que si quiere MATARAPPAUL necesitara tener en su poder un fuerte EXPLOSIVO, por lo que quiere conseguirlo antes de ir a por PAUL SABAL dispara TENIENTEPAUL con su MP44 en BANAPUR

El ruin YUMA quiere el potente ANTIDOTO para salvar su vida

En la TORRE de TIRTHA, AMITA utiliza su MP5 para acabar con un soldado

SABAL dispara TENIENTEYUMA con su MP44 en BANAPUR

En la TORRE de TIRTHA, AMITA utiliza su MP5 para acabar con un soldado

En TIRTHA, AMITA consigue que la TORRE quede DESTRUIDA completando así su misión DESTRUIRLATORRE

TENIENTEPAUL dispara SABAL con su AK-74 en BANAPUR

En la TORRE de TIRTHA, AMITA coge un mortífero ANTIDOTO

AJAY se dirige de BANAPUR a la IGLESIA de VARSHAKOT para llevar a cabo su misión LIBERARLAIGLESIA

TENIENTEPAGAN se dirige de FORTALEZA_PAGAN a BANAPUR para atacar a un miembro de la Senda Dorada

TENIENTEPAGAN dispara SABAL con su MINIUZI, SABAL no aguanta más disparos y muere en BANAPUR

AMITA desea recuperar la cultura POLITICA de Kirat, por lo que va a intentar DESTRUIRLAFABRICA de Kirat, que controla la Guardia Real

AMITA se dirige de TIRTHA a la FABRICA de SHANATH para llevar a cabo su misión DESTRUIRLAFABRICA

En la FABRICA de SHANATH, AMITA utiliza su MP5 para acabar con un soldado

TENIENTEPAUL se dirige de BANAPUR a SHANATH para atacar a un miembro de la Senda Dorada

En la FABRICA de SHANATH. TENIENTEPAUL coge un formidable ARMAMENTO

TENIENTEPAGAN se dirige de BANAPUR a SHANATH porque quiere ARMAMENTO que lo tiene TENIENTEPAUL

TENIENTEYUMA se dirige de BANAPUR a SHANATH para atacar a un miembro de la Senda Dorada

El miserable TENIENTEYUMA quiere conseguir el ANTIDOTO que podría acabar con su líder YUMA

AMITA dispara TENIENTEYUMA con su MP5, TENIENTEYUMA no aguanta más disparos y muere en SHANATH

AMITA dispara TENIENTEPAUL con su MP5, TENIENTEPAUL no aguanta más disparos y muere en SHANATH

En SHANATH, AMITA coge ARMAMENTO del cuerpo sin vida de TENIENTEPAUL

En la IGLESIA de VARSHAKOT, AJAY utiliza C4 para acabar con todos los soldados

En la FABRICA de SHANATH, AMITA utiliza su MP5 para acabar con un soldado

AMITA dispara TENIENTEPAGAN con su MP5 en SHANATH

En VARSHAKOT, AJAY consigue que la IGLESIA quede LIBERADA completando así su misión LIBERARLAIGLESIA

AMITA consulta a AJAY si sería capaz de MATARAPPAUL, uno de los líderes que tiene oprimido al pueblo de Kirat, AJAY sin dudarlo se embarca en la misión

En la IGLESIA de VARSHAKOT, AJAY coge un mortífero EXPLOSIVO

En SHANATH, AMITA consigue que la FABRICA quede DESTRUIDA completando así su misión DESTRUIRLAFABRICA

AMITA cansada de la opresión bajo la que se encuentran los ciudadanos de Kirat, va a intentar MATARAYUMA acabando así con uno de los líderes de la dictadura
AJAY se dirige de VARSHAKOT a FORTALEZA_PAUL con el letal EXPLOSIVO para acabar con PAUL

AMITA dispara TENIENTEPAGAN con su MP5, TENIENTEPAGAN no aguanta más disparos y muere en SHANATH

AMITA se dirige de SHANATH a FORTALEZA_YUMA con el letal ANTIDOTO para acabar con YUMA

En FORTALEZA_PAUL, AJAY mata PAUL utilizando el gran EXPLOSIVO

En FORTALEZA_YUMA, AMITA mata YUMA utilizando el gran ANTIDOTO

AMITA cansada de la opresión bajo la que se encuentran los ciudadanos de Kirat, va a intentar MATARAPAGAN acabando así con uno de los líderes de la dictadura

AMITA se dirige de FORTALEZA_YUMA a FORTALEZA_PAGAN con el letal ARMAMENTO para acabar con PAGAN

En FORTALEZA_PAGAN, AMITA mata PAGAN utilizando el gran ARMAMENTO

End.

Datos Obtenidos para el Problema 4

aleatoriedad (0,001%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	Si	Si	Si	Si	-
Longitud	43	43	43	43	43
Diversidad	-	0	0	0	0
aleatoriedad (20%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	Si	Si	Si	Si	-
Longitud	45	47	51	49	48
Diversidad	-	22	60	68	50
aleatoriedad (50%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	Si	No	No	No	-
Longitud	50	40	46	52	-
Diversidad	-	72	62	68	-
aleatoriedad (70%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	No	No	No	Si	-
Longitud	46	36	43	51	-
Diversidad	-	66	61	68	-

Tabla 85: Datos Problema 4

6.2.5. Problema 5

Definición

La definición del problema 5 es:

```
(:goal (and
      (mision-complete SalvarAPaul)
      (mision-complete SalvarAYuma)
      (mision-complete SalvarAPagan)
    )
)
```

En este problema se van a generar planes en los que los personajes de la Guardia tienen que completar sus misiones “SalvarPaul”, “SalvarYuma” y “SalvarPagan”. Para completar estas misiones los personajes de la Guardia Real tendrán que llevar a cabo las acciones oportunas.

Resultado

A continuación, se muestra uno de los planes encontrados por el planificador para el problema 5. En concreto este problema se ha obtenido aplicando una probabilidad de aleatoriedad del 20%.

Plan 1

El ruin PAGAN quiere el potente ARMAMENTO para salvar su vida

El ruin YUMA quiere el potente ANTIDOTO para salvar su vida

El ruin PAUL quiere el potente EXPLOSIVO para salvar su vida

El miserable TENIENTEYUMA quiere conseguir el ANTIDOTO que podría acabar con su líder YUMA

El miserable TENIENTEPAUL quiere conseguir el EXPLOSIVO que podría acabar con su líder PAUL

AMITA consulta a AJAY si sería capaz de MATARAYUMA, uno de los líderes que tiene oprimido al pueblo de Kirat, AJAY sin dudarlo se embarca en la misión

El miserable TENIENTEPAGAN quiere conseguir el ARMAMENTO que podría acabar con su líder PAGAN

TENIENTEPAGAN se dirige de FORTALEZA_PAGAN a BANAPUR para atacar a un miembro de la Senda Dorada

AMITA desea recuperar la cultura POLITICA de Kirat, por lo que va a intentar DESTRUIRLATORRE de Kirat, que controla la Guardia Real

SABAL desea recuperar la cultura RELIGIOSA de Kirat, por lo que va a intentar LIBERARLAIGLESIA de Kirat, que controla la Guardia Real

SABAL se dirige de BANAPUR a la IGLESIA de VARSHAKOT para llevar a cabo su misión LIBERARLAIGLESIA

TENIENTEPAUL se dirige de FORTALEZA_PAUL a BANAPUR para atacar a un miembro de la Senda Dorada

TENIENTEYUMA se dirige de FORTALEZA_YUMA a la TORRE de TIRTHA dado que allí se sitúa el eficaz ANTIDOTO que necesita

En la TORRE de TIRTHA, TENIENTEYUMA coge un formidable ANTIDOTO

TENIENTEYUMA se dirige de TIRTHA a FORTALEZA_YUMA porque quiere darle ANTIDOTO a su líder YUMA

En FORTALEZA_YUMA, TENIENTEYUMA le da ANTIDOTO a su líder YUMA

TENIENTEPAGAN dispara AJAY con su MINIUZI en BANAPUR

YUMA consigue salvar su vida destruyendo el abrumador ANTIDOTO. Destruído el artefacto, la Senda Dorada ya no podrá acabar con el

TENIENTEPAGAN se dirige de BANAPUR a la FABRICA de SHANATH dado que allí se sitúa el eficaz ARMAMENTO que necesita

En la FABRICA de SHANATH, TENIENTEPAGAN coge un formidable ARMAMENTO

TENIENTEPAGAN se dirige de SHANATH a FORTALEZA_PAGAN porque quiere darle ARMAMENTO a su líder PAGAN

En FORTALEZA_PAGAN, TENIENTEPAGAN le da ARMAMENTO a su líder PAGAN

PAGAN consigue salvar su vida destruyendo el abrumador ARMAMENTO. Destruído el artefacto, la Senda Dorada ya no podrá acabar con el

TENIENTEYUMA se dirige de FORTALEZA_YUMA a VARSHAKOT para atacar a un miembro de la Senda Dorada

En la IGLESIA de VARSHAKOT, TENIENTEYUMA coge un formidable EXPLOSIVO

SABAL dispara TENIENTEYUMA con su MP44 en VARSHAKOT

TENIENTEYUMA dispara SABAL con su G36 en VARSHAKOT

TENIENTEYUMA se dirige de VARSHAKOT a FORTALEZA_PAUL porque quiere darle EXPLOSIVO a su líder PAUL

En FORTALEZA_PAUL, TENIENTEYUMA le da EXPLOSIVO a su líder PAUL

PAUL consigue salvar su vida destruyendo el abrumador EXPLOSIVO. Destruído el artefacto, la Senda Dorada ya no podrá acabar con el

End.

Datos Obtenidos para el Problema 5

aleatoriedad (0,001%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	Si	Si	Si	Si	-
Longitud	21	21	21	21	21
Diversidad	-	0	0	0	0
aleatoriedad (20%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	Si	Si	Si	Si	-
Longitud	30	41	39	50	40
Diversidad	-	28	54	63	48
aleatoriedad (50%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	Si	Si	Si	Si	-
Longitud	60	55	34	41	47
Diversidad	-	95	61	43	66
aleatoriedad (70%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	No	Si	No	Si	-
Longitud	42	52	43	53	52
Diversidad	-	78	77	76	-

Tabla 86: Datos Problema 5

6.2.6. Problema 6

Definición

La definición del problema 6 es:

```
(:goal (and
      (mision-complete SalvarAPaul)
      (mision-complete SalvarAYuma)
      (mision-complete SalvarAPagan)
      ;+
      (mision-complete LiberarLaIglesia)
      (mision-complete DestruirLaFabrica)
      (mision-complete DestruirLaTorre)
    )
)
```

En este problema se parte de las metas definidas en el problema 5 pero además se añade que los personajes de la Senda Dorada tienen que realizar las misiones de “LiberarLaIglesia”, “DestruirLaFabrica” y “DestruirLaTorre”. Estas misiones que ya fueron probadas en el problema 1 se van a probar en este problema para ver si se siguen completando con éxito cuando los personajes de la Guardia Real también completan sus misiones.

Resultado

A continuación, se muestra uno de los planes encontrados por el planificador para el problema 6. En concreto este problema se ha obtenido aplicando una probabilidad de aleatoriedad del 20%.

Plan 1

SABAL le pide a AJAY que le ayude a LIBERARLAIGLESIA de Kirat, AJAY decidido a ayudar a su compañero a recuperar la cultura RELIGIOSA de kirat acepta la misión
El ruin PAGAN quiere el potente ARMAMENTO para salvar su vida
El ruin YUMA quiere el potente ANTIDOTO para salvar su vida
El ruin PAUL quiere el potente EXPLOSIVO para salvar su vida
El miserable TENIENTEYUMA quiere conseguir el ANTIDOTO que podría acabar con su líder YUMA
El miserable TENIENTEPAUL quiere conseguir el EXPLOSIVO que podría acabar con su líder PAUL
El miserable TENIENTEPAGAN quiere conseguir el ARMAMENTO que podría acabar con su líder PAGAN
AJAY se dirige de BANAPUR a la IGLESIA de VARSHAKOT para llevar a cabo su misión LIBERARLAIGLESIA
AMITA desea recuperar la cultura POLITICA de Kirat, por lo que va a intentar DESTRUIRLATORRE de Kirat, que controla la Guardia Real
SABAL cansado de la opresión bajo la que se encuentran los ciudadanos de Kirat, va a intentar MATARAPAGAN acabando así con uno de los líderes de la dictadura

SABAL sabe que si quiere MATARAPAGAN necesitara tener en su poder un fuerte ARMAMENTO, por lo que quiere conseguirlo antes de ir a por PAGAN

En la IGLESIA de VARSHAKOT, AJAY utiliza C4 para acabar con todos los soldados
AMITA se dirige de BANAPUR a la TORRE de TIRTHA para llevar a cabo su misión DESTRUIRLATORRE

En la TORRE de TIRTHA, AMITA utiliza su MP5 para acabar con un soldado

En la TORRE de TIRTHA, AMITA utiliza su MP5 para acabar con un soldado

SABAL se dirige de BANAPUR a la FABRICA de SHANATH dado que allí se sitúa el eficaz ARMAMENTO que necesita

En la FABRICA de SHANATH, SABAL utiliza su MP44 para acabar con un soldado

En la FABRICA de SHANATH, SABAL utiliza su MP44 para acabar con un soldado

En VARSHAKOT, AJAY consigue que la IGLESIA quede LIBERADA completando así su misión LIBERARLAIGLESIA

AMITA consulta a AJAY si sería capaz de MATARAYUMA, uno de los líderes que tiene oprimido al pueblo de Kirat, AJAY sin dudarlo se embarca en la misión

En TIRTHA, AMITA consigue que la TORRE quede DESTRUIDA completando así su misión DESTRUIRLATORRE

TENIENTEPAUL se dirige de FORTALEZA_PAUL a SHANATH para atacar a un miembro de la Senda Dorada

AMITA desea recuperar la cultura POLITICA de Kirat, por lo que va a intentar DESTRUIRLAFABRICA de Kirat, que controla la Guardia Real

TENIENTEPAGAN se dirige de FORTALEZA_PAGAN a SHANATH para atacar a un miembro de la Senda Dorada

En la FABRICA de SHANATH, TENIENTEPAUL coge un formidable ARMAMENTO

En la TORRE de TIRTHA, AMITA coge un mortífero ANTIDOTO

TENIENTEPAUL dispara SABAL con su AK-74 en SHANATH

TENIENTEPAUL se dirige de SHANATH a la IGLESIA de VARSHAKOT dado que allí se sitúa el eficaz EXPLOSIVO que necesita

En la IGLESIA de VARSHAKOT, TENIENTEPAUL coge un formidable EXPLOSIVO

En FABRICA, SABAL consigue que la FABRICA quede DESTRUIDA completando así la misión DESTRUIRLAFABRICA que quería realizar su compañero

TENIENTEPAUL se dirige de VARSHAKOT a FORTALEZA_PAGAN porque quiere darle ARMAMENTO a su líder PAGAN

En FORTALEZA_PAGAN, TENIENTEPAUL le da ARMAMENTO a su líder PAGAN

AMITA cansada de la opresión bajo la que se encuentran los ciudadanos de Kirat, va a intentar MATARAPAGAN acabando así con uno de los líderes de la dictadura

PAGAN consigue salvar su vida destruyendo el abrumador ARMAMENTO. Destruído el artefacto, la Senda Dorada ya no podrá acabar con el

TENIENTEPAUL se dirige de FORTALEZA_PAGAN a FORTALEZA_PAUL porque quiere darle EXPLOSIVO a su líder PAUL

En FORTALEZA_PAUL, TENIENTEPAUL le da EXPLOSIVO a su líder PAUL

PAUL consigue salvar su vida destruyendo el abrumador EXPLOSIVO. Destruído el artefacto, la Senda Dorada ya no podrá acabar con el

TENIENTEYUMA se dirige de FORTALEZA_YUMA a TIRTHA porque quiere ANTIDOTO que lo tiene AMITA

TENIENTEYUMA dispara AMITA con su G36 en TIRTHA

TENIENTEYUMA dispara AMITA con su G36, AMITA no aguanta más disparos y muere en TIRTHA

En TIRTHA, TENIENTEYUMA coge ANTIDOTO del cuerpo sin vida de AMITA

TENIENTEYUMA se dirige de TIRTHA a FORTALEZA_YUMA porque quiere darle ANTIDOTO a su líder YUMA

En FORTALEZA_YUMA, TENIENTEYUMA le da ANTIDOTO a su líder YUMA

TENIENTEPAGAN dispara SABAL con su MINIUZI, SABAL no aguanta más disparos y muere en SHANATH

*YUMA consigue salvar su vida destruyendo el abrumador ANTIDOTO. Destruído el artefacto, la Senda Dorada ya no podrá acabar con el
End.*

Datos Obtenidos para el Problema 6

aleatoriedad (0,001%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	Si	Si	Si	Si	-
Longitud	36	36	36	36	36
Diversidad	-	-		-	0
aleatoriedad (20%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	Si	Si	Si	Si	-
Longitud	45	43	57	41	46
Diversidad	-	26	70	72	56
aleatoriedad (50%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	No	No	No	Si	-
Longitud	51	55	57	56	-
Diversidad	-	76	86	77	-
aleatoriedad (70%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	No	No	No	No	-
Longitud	19	46	45	61	-
Diversidad	-	51	61	64	-

Tabla 87: Datos Problema 6

6.2.7. Problema 7

Definición

La definición del problema 7 es:

```
(:goal (and
  (mision-complete SalvarAPaul)
  (mision-complete SalvarAYuma)
  (mision-complete SalvarAPagan)
  ;+
  (mision-complete LiberarLaIglesia)
  (mision-complete DestruirLaFabrica)
  (mision-complete DestruirLaTorre)
  ;+
  (personaje-muerto Sabal)
  (personaje-muerto Ajay)
  (personaje-muerto Amita)
)
```

El problema 7 no es más que una variación de la definición del problema 6 a la que se añade que los 3 personajes de la Senda Dorada tienen que acabar muertos. Es un problema interesante ya que los personajes de la Senda Dorada deberán completar algunas de sus misiones antes de que los tres mueran.

Resultado.

A continuación, se muestra uno de los planes encontrados por el planificador para el problema 7. En concreto este problema se ha obtenido aplicando una probabilidad de aleatoriedad del 20%.

Plan 1

SABAL le pide a AJAY que le ayude a LIBERARLAIGLESIA de Kirat, AJAY decidido a ayudar a su compañero a recuperar la cultura RELIGIOSA de Kirat acepta la misión
El ruin PAGAN quiere el potente ARMAMENTO para salvar su vida
El ruin YUMA quiere el potente ANTIDOTO para salvar su vida
El ruin PAUL quiere el potente EXPLOSIVO para salvar su vida
AMITA desea recuperar la cultura POLITICA de Kirat, por lo que va a intentar DESTRUIRLATORRE de Kirat, que controla la Guardia Real
SABAL cansado de la opresión bajo la que se encuentran los ciudadanos de Kirat, va a intentar MATARAPAGAN acabando así con uno de los líderes de la dictadura
SABAL sabe que si quiere MATARAPAGAN necesitara tener en su poder un fuerte ARMAMENTO, por lo que quiere conseguirlo antes de ir a por PAGAN
El miserable TENIENTEYUMA quiere conseguir el ANTIDOTO que podría acabar con su líder YUMA

El miserable TENIENTEPAUL quiere conseguir el EXPLOSIVO que podría acabar con su líder PAUL

El miserable TENIENTEPAGAN quiere conseguir el ARMAMENTO que podría acabar con su líder PAGAN

TENIENTEYUMA se dirige de FORTALEZA_YUMA a BANAPUR para atacar a un miembro de la Senda Dorada

TENIENTEPAUL se dirige de FORTALEZA_PAUL a la IGLESIA de VARSHAKOT dado que allí se sitúa el eficaz EXPLOSIVO que necesita

En la IGLESIA de VARSHAKOT. TENIENTEPAUL coge un formidable EXPLOSIVO

TENIENTEPAUL se dirige de VARSHAKOT a FORTALEZA_PAUL porque quiere darle EXPLOSIVO a su líder PAUL

TENIENTEPAUL se dirige de FORTALEZA_PAUL a BANAPUR para atacar a un miembro de la Senda Dorada

AMITA se dirige de BANAPUR a la TORRE de TIRTHA para llevar a cabo su misión DESTRUIRLATORRE

SABAL dispara TENIENTEYUMA con su MP44 en BANAPUR

En la TORRE de TIRTHA, AMITA utiliza su MP5 para acabar con un soldado

En la TORRE de TIRTHA, AMITA utiliza su MP5 para acabar con un soldado

SABAL se dirige de BANAPUR a la FABRICA de SHANATH dado que allí se sitúa el eficaz ARMAMENTO que necesita

TENIENTEPAGAN se dirige de FORTALEZA_PAGAN a TIRTHA para atacar a un miembro de la Senda Dorada

En la TORRE de TIRTHA . TENIENTEPAGAN coge un formidable ANTIDOTO

En la FABRICA de SHANATH, SABAL utiliza su MP44 para acabar con un soldado

En la FABRICA de SHANATH, SABAL utiliza su MP44 para acabar con un soldado

En SHANATH, SABAL consigue que la FABRICA quede DESTRUIDA completando así la misión de DESTRUIRLAFABRICA

En la FABRICA de SHANATH. SABAL coge un mortífero ARMAMENTO

En TIRTHA, AMITA consigue que la TORRE quede DESTRUIDA completando así su misión DESTRUIRLATORRE

AJAY dispara TENIENTEPAUL con su AK-47 en BANAPUR

SABAL se dirige de SHANATH a FORTALEZA_PAGAN con el letal ARMAMENTO para acabar con PAGAN

PAGAN dispara SABAL con su M16 en FORTALEZA_PAGAN

TENIENTEYUMA dispara AJAY con su G36 en BANAPUR

TENIENTEPAGAN dispara AMITA con su MINIUZI en TIRTHA

PAGAN dispara SABAL con su M16, SABAL no aguanta más disparos y muere en FORTALEZA_PAGAN

En FORTALEZA_PAGAN, PAGAN coge ARMAMENTO del cuerpo sin vida de SABAL

PAGAN consigue salvar su vida destruyendo el abrumador ARMAMENTO. Destruído el artefacto, la Senda Dorada ya no podrá acabar con el

TENIENTEYUMA dispara AJAY con su G36, AJAY no aguanta más disparos y muere en BANAPUR

AMITA triste por la muerte de su camarada SABAL intentara cumplir su deseo de recuperar la cultura Religiosa de Kirat

AMITA desea recuperar la cultura RELIGIOSA de Kirat, por lo que va a intentar LIBERARLAIGLESIA de Kirat, que controla la Guardia Real

TENIENTEPAUL se dirige de BANAPUR a FORTALEZA_PAUL porque quiere darle EXPLOSIVO a su líder PAUL

En FORTALEZA_PAUL, TENIENTEPAUL le da EXPLOSIVO a su líder PAUL

PAUL consigue salvar su vida destruyendo el abrumador EXPLOSIVO. Destruído el artefacto, la Senda Dorada ya no podrá acabar con el

AMITA se dirige de TIRTHA a la IGLESIA de VARSHAKOT para llevar a cabo su misión LIBERARLAIGLESIA

En la IGLESIA de VARSHAKOT, AMITA utiliza su MP5 para acabar con un soldado

En la IGLESIA de VARSHAKOT, AMITA utiliza su MP5 para acabar con un soldado

En VARSHAKOT, AMITA consigue que la IGLESIA quede LIBERADA completando así su misión LIBERARLAIGLESIA

TENIENTEPAGAN se dirige de TIRTHA a FORTALEZA_YUMA porque quiere darle ANTIDOTO a su líder YUMA

AMITA cansada de la opresión bajo la que se encuentran los ciudadanos de Kirat, va a intentar MATARAPPAUL acabando así con uno de los líderes de la dictadura

En FORTALEZA_YUMA, TENIENTEPAGAN le da ANTIDOTO a su líder YUMA

YUMA consigue salvar su vida destruyendo el abrumador ANTIDOTO. Destruído el artefacto, la Senda Dorada ya no podrá acabar con el

TENIENTEPAUL se dirige de FORTALEZA_PAUL a VARSHAKOT para atacar a un miembro de la Senda Dorada

TENIENTEPAGAN se dirige de FORTALEZA_YUMA a VARSHAKOT para atacar a un miembro de la Senda Dorada

AMITA dispara TENIENTEPAUL con su MP5, TENIENTEPAUL no aguanta más disparos y muere en VARSHAKOT

TENIENTEPAGAN dispara AMITA con su MINIUZI, AMITA no aguanta más disparos y muere en VARSHAKOT

End.

Datos Obtenidos para el Problema 7

aleatoriedad (0,001%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	Si	Si	Si	Si	-
Longitud	42	42	42	42	42
Diversidad	-	0	0	0	0
aleatoriedad (20%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	Si	Si	Si	Si	-
Longitud	47	51	53	48	50
Diversidad	-	68	82	43	64
aleatoriedad (50%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	No	No	No	Si	-
Longitud	53	25	43	54	-
Diversidad	-	56	46	51	-
aleatoriedad (70%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	No	No	No	No	-
Longitud	41	48	48	42	-
Diversidad	-	67	66	68	-

Tabla 88: Datos Problema 7

6.2.8. Problema 8

Definición

La definición del problema 8 es:

```
(:goal (and
  (mision-complete SalvarAPaul)
  (mision-complete MatarAPagan)
  (mision-complete MatarAYuma)
  ;+
  (mision-complete LiberarLaIglesia)
  (mision-complete DestruirLaTorre)
  (mision-complete DestruirLaFabrica)
  ;+
  (personaje-muerto TenientePagan)
)
```

El problema 8 mezcla las misiones de “SalvarA” de la Guardia Real con las de “MatarA” de la Senda Dorada. En este problema se debe completar la misión de “SalvarAPaul” “MatarAPagan” y “MatarAYuma”. Además, se deben completar las misiones básicas de la Senda Dorada que tienen que ver con liberar o destruir las construcciones. Por último, para completar el problema el TenientePagan debe morir. Este problema resulta interesante porque en él se mezclan distintos objetivos.

Resultado

A continuación, se muestra uno de los planes encontrados por el planificador para el problema 8. En concreto este problema se ha obtenido aplicando una probabilidad de aleatoriedad del 20%.

Plan 1

AMITA le pide a AJAY que le ayude a DESTRUIRLATORRE de Kirat, AJAY decidido a ayudar a su compañero a recuperar la cultura POLITICA de Kirat acepta la misión
SABAL desea recuperar la cultura RELIGIOSA de Kirat, por lo que va a intentar LIBERARLAIGLESIA de Kirat, que controla la Guardia Real
AMITA desea recuperar la cultura POLITICA de Kirat, por lo que va a intentar DESTRUIRLAFABRICA de Kirat, que controla la Guardia Real
El ruin PAUL quiere el potente EXPLOSIVO para salvar su vida
El miserable TENIENTEPAUL quiere conseguir el EXPLOSIVO que podría acabar con su líder PAUL
El miserable TENIENTEPAGAN quiere conseguir el ARMAMENTO que podría acabar con su líder PAGAN
TENIENTEPAGAN se dirige de FORTALEZA_PAGAN a la FABRICA de SHANATH dado que allí se sitúa el eficaz ARMAMENTO que necesita
En la FABRICA de SHANATH. TENIENTEPAGAN coge un formidable ARMAMENTO

AMITA se dirige de BANAPUR a la FABRICA de SHANATH para llevar a cabo su misión DESTRUIRLAFABRICA

AJAY se dirige de BANAPUR a la TORRE de TIRTHA para llevar a cabo su misión DESTRUIRLATORRE

En la FABRICA de SHANATH, AMITA utiliza su MP5 para acabar con un soldado

SABAL se dirige de BANAPUR a la IGLESIA de VARSHAKOT para llevar a cabo su misión LIBERARLAIGLESIA

TENIENTEPAGAN se dirige de SHANATH a TIRTHA para atacar a un miembro de la Senda Dorada

El ruin PAGAN quiere el potente ARMAMENTO para salvar su vida

TENIENTEPAUL se dirige de FORTALEZA_PAUL a SHANATH para atacar a un miembro de la Senda Dorada

En la IGLESIA de VARSHAKOT, SABAL utiliza su MP44 para acabar con un soldado

En la TORRE de TIRTHA, TENIENTEPAGAN coge un formidable ANTIDOTO

En la TORRE de TIRTHA, AJAY utiliza C4 para acabar con todos los soldados

El miserable TENIENTEYUMA quiere conseguir el ANTIDOTO que podría acabar con su líder YUMA

En la FABRICA de SHANATH, AMITA utiliza su MP5 para acabar con un soldado

En TIRTHA, AJAY consigue que la TORRE quede DESTRUIDA completando así su misión DESTRUIRLATORRE

En la IGLESIA de VARSHAKOT, SABAL utiliza su MP44 para acabar con un soldado

En SHANATH, AMITA consigue que la FABRICA quede DESTRUIDA completando así su misión DESTRUIRLAFABRICA

TENIENTEYUMA se dirige de FORTALEZA_YUMA a VARSHAKOT para atacar a un miembro de la Senda Dorada

El ruin YUMA quiere el potente ANTIDOTO para salvar su vida

En la IGLESIA de VARSHAKOT, TENIENTEYUMA coge un formidable EXPLOSIVOAJAY dispara TENIENTEPAGAN con su AK-47 en TIRTHA

En VARSHAKOT, SABAL consigue que la IGLESIA quede LIBERADA completando así su misión LIBERARLAIGLESIA

AJAY dispara TENIENTEPAGAN con su AK-47, TENIENTEPAGAN no aguanta más disparos y muere en TIRTHA

En TIRTHA, AJAY coge ANTIDOTO del cuerpo sin vida de TENIENTEPAGAN

SABAL consulta a AJAY si sería capaz de MATARAPAGAN, uno de los líderes que tiene oprimido al pueblo de Kirat, AJAY sin dudarlo se embarca en la misión

En TIRTHA, AJAY coge ARMAMENTO del cuerpo sin vida de TENIENTEPAGAN

AJAY se dirige de TIRTHA a FORTALEZA_PAGAN con el letal ARMAMENTO para acabar con PAGAN

En FORTALEZA_PAGAN, AJAY mata PAGAN utilizando el gran ARMAMENTO

SABAL consulta a AJAY si sería capaz de MATARAYUMA, uno de los líderes que tiene oprimido al pueblo de Kirat, AJAY sin dudarlo se embarca en la misión

AJAY se dirige de FORTALEZA_PAGAN a FORTALEZA_YUMA con el letal ANTIDOTO para acabar con YUMA

En FORTALEZA_YUMA, AJAY mata YUMA utilizando el gran ANTIDOTO

SABAL cansado de la opresión bajo la que se encuentran los ciudadanos de Kirat, va a intentar MATARAPPAUL acabando así con uno de los líderes de la dictadura

SABAL dispara TENIENTEYUMA con su MP44 en VARSHAKOT

TENIENTEYUMA dispara SABAL con su G36 en VARSHAKOT

TENIENTEYUMA se dirige de VARSHAKOT a FORTALEZA_PAUL porque quiere darle EXPLOSIVO a su líder PAUL

En FORTALEZA_PAUL, TENIENTEYUMA le da EXPLOSIVO a su líder PAUL

TENIENTEYUMA se dirige de FORTALEZA_PAUL a SHANATH para atacar a un miembro de la Senda Dorada

PAUL consigue salvar su vida destruyendo el abrumador EXPLOSIVO. Destruído el artefacto, la Senda Dorada ya no podrá acabar con el

End.

Datos Obtenidos para el Problema 8

aleatoriedad (0,001%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	Si	Si	Si	Si	-
Longitud	36	36	36	36	36
Diversidad	-	0	0	0	0
aleatoriedad (20%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	Si	Si	Si	Si	-
Longitud	49	44	51	46	39
Diversidad	-	43	49	55	49
aleatoriedad (50%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	Si	No	No	No	-
Longitud	45	45	46	60	-
Diversidad	-	68	71	94	-
aleatoriedad (70%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	No	No	No	No	-
Longitud	41	58	54	49	
Diversidad	-	59	86	73	-

Tabla 89: Datos Problema 8

6.2.9. Problema 9

Definición

La definición del problema 9 es:

```
(:goal (and
  (mision-complete SalvarAPaul)
  (mision-complete SalvarAYuma)
  (mision-complete MatarPagan)
  ;+
  (mision-complete LiberarLaIglesia)
  (mision-complete DestruirLaTorre)
  (mision-complete DestruirLaFabrica)
  ;+
  (asesinado-por TenienteYuma Sabal)
  ;+
  (asesinado-por Ajay Pagan)
)
```

Este problema es muy parecido al anterior ya que mezcla distintas misiones, pero se añade una dificultad. En este problema además de completarse misiones del mismo tipo de las que se completan en el problema anterior los personajes Ajay y TenienteYuma deben acabar muertos. Pero no les pueden matar cualquier personaje, sino que les deben abatir Pagan y Sabal respectivamente.

Resultado

A continuación, se muestra uno de los planes encontrados por el planificador para el problema 9. En concreto este problema se ha obtenido aplicando una probabilidad de aleatoriedad del 20%.

Plan 1

SABAL le pide a AJAY que le ayude a LIBERARLAIGLESIA de Kirat, AJAY decidido a ayudar a su compañero a recuperar la cultura RELIGIOSA de Kirat acepta la misión
AMITA desea recuperar la cultura POLITICA de Kirat, por lo que va a intentar DESTRUIRLAFABRICA de Kirat, que controla la Guardia Real
El ruin PAGAN quiere el potente ARMAMENTO para salvar su vida
El ruin PAUL quiere el potente EXPLOSIVO para salvar su vida
SABAL cansado de la opresión bajo la que se encuentran los ciudadanos de Kirat, va a intentar MATARAYUMA acabando así con uno de los líderes de la dictadura
SABAL sabe que si quiere MATARAYUMA necesitara tener en su poder un fuerte ANTIDOTO, por lo que quiere conseguirlo antes de ir a por YUMA
El miserable TENIENTEYUMA quiere conseguir el ANTIDOTO que podría acabar con su líder YUMA
El miserable TENIENTEPAUL quiere conseguir el EXPLOSIVO que podría acabar con su líder PAUL

El miserable TENIENTEPAGAN quiere conseguir el ARMAMENTO que podría acabar con su líder PAGAN

TENIENTEPAGAN se dirige de FORTALEZA_PAGAN a BANAPUR para atacar a un miembro de la Senda Dorada

AJAY se dirige de BANAPUR a la IGLESIA de VARSHAKOT para llevar a cabo su misión LIBERARLAIGLESIA

En la IGLESIA de VARSHAKOT, AJAY utiliza C4 para acabar con todos los soldados

En VARSHAKOT, AJAY consigue que la IGLESIA quede LIBERADA completando así su misión LIBERARLAIGLESIA

AMITA consulta a AJAY si sería capaz de MATARAPAGAN, uno de los líderes que tiene oprimido al pueblo de Kirat, AJAY sin dudarlo se embarca en la misión

AJAY sabe que si quiere MATARAPAGAN necesitara tener en su poder un fuerte ARMAMENTO, por lo que quiere conseguirlo antes de ir a por PAGAN

SABAL se dirige de BANAPUR a la TORRE de TIRTHA dado que allí se sitúa el eficaz ANTIDOTO que necesita

En la TORRE de TIRTHA, SABAL utiliza su MP44 para acabar con un soldado

En la TORRE de TIRTHA, SABAL utiliza su MP44 para acabar con un soldado

En TIRTHA, SABAL consigue que la TORRE quede DESTRUIDA completando así la misión de DESTRUIRLATORRE

En la TORRE de TIRTHA. SABAL coge un mortífero ANTIDOTO

TENIENTEPAUL se dirige de FORTALEZA_PAUL a la IGLESIA de VARSHAKOT dado que allí se sitúa el eficaz EXPLOSIVO que necesita

En la IGLESIA de VARSHAKOT. TENIENTEPAUL coge un formidable EXPLOSIVO

AJAY se dirige de VARSHAKOT a la FABRICA de SHANATH dado que allí se sitúa el eficaz ARMAMENTO que necesita

TENIENTEPAUL se dirige de VARSHAKOT a TIRTHA para atacar a un miembro de la Senda Dorada

En la FABRICA de SHANATH, AJAY utiliza C4 para acabar con todos los soldados

En SHANATH, AJAY consigue que la FABRICA quede DESTRUIDA completando así la misión DESTRUIRLAFABRICA que quería realizar su compañero

En la FABRICA de SHANATH. AJAY coge un mortífero ARMAMENTO

SABAL se dirige de TIRTHA a FORTALEZA_YUMA con el letal ANTIDOTO para acabar con YUMA

AMITA dispara TENIENTEPAGAN con su MP5 en BANAPUR

TENIENTEPAUL se dirige de TIRTHA a FORTALEZA_PAUL porque quiere darle EXPLOSIVO a su líder PAUL

TENIENTEPAUL se dirige de FORTALEZA_PAUL a SHANATH para atacar a un miembro de la Senda Dorada

AJAY se dirige de SHANATH a FORTALEZA_PAGAN con el letal ARMAMENTO para acabar con PAGAN

TENIENTEPAUL se dirige de SHANATH a FORTALEZA_PAUL porque quiere darle EXPLOSIVO a su líder PAUL

En FORTALEZA_PAUL, TENIENTEPAUL le da EXPLOSIVO a su líder PAUL

PAUL consigue salvar su vida destruyendo el abrumador EXPLOSIVO. Destruído el artefacto, la Senda Dorada ya no podrá acabar con el

PAGAN dispara AJAY con su M16 en FORTALEZA_PAGAN

SABAL dispara TENIENTEYUMA con su MP44 en FORTALEZA_YUMA

PAGAN dispara AJAY con su M16, AJAY no aguanta más disparos y muere en FORTALEZA_PAGAN

En FORTALEZA_PAGAN, PAGAN coge ARMAMENTO del cuerpo sin vida de AJAY

PAGAN consigue salvar su vida destruyendo el abrumador ARMAMENTO. Destruído el artefacto, la Senda Dorada ya no podrá acabar con el

SABAL dispara TENIENTEYUMA con su MP44, TENIENTEYUMA no aguanta más disparos y muere en FORTALEZA_YUMA

En FORTALEZA_YUMA, SABAL mata YUMA utilizando el gran ANTIDOTO

End

Datos Obtenidos para el Problema 9

aleatoriedad (0,001%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	Si	Si	Si	Si	-
Longitud	42	42	42	42	42
Diversidad	-	0	0	0	0
aleatoriedad (20%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	No	Si	No	No	-
Longitud	50	42	37	37	-
Diversidad	-	74	63	52	-
aleatoriedad (50%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	No	No	No	No	-
Longitud	19	46	44	30	-
Diversidad	-	45	66	60	-
aleatoriedad (70%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	No	No	No	No	-
Longitud	45	31	34	33	-
Diversidad	-	50	47	55	-

Tabla 90: Datos Problema 9

6.2.10. Problema 10

Definición

La definición del problema 10 es:

(:goal (and

```
(mision-complete SalvarAPaul)
(mision-complete SalvarAPagan)
(mision-complete MatarAYuma)
;+
(mision-complete LiberarLaIglesia)
(mision-complete DestruirLaTorre)
(mision-complete DestruirLaFabrica)
;+
(asesinado-por TenienteYuma Sabal)
(asesinado-por TenientePaul Ajay)
;+
(asesinado-por Ajay Pagan)
;+
(personaje-vida-quitada Amita zero)
(personaje-vida-quitada Sabal zero)
(personaje-vida-quitada TenientePagan zero)
```

)
)

El ultimo problema que se va a probar es muy parecido al problema 9 pero en este problema además de mezclar distintas misiones y que un personaje sea matado por otro, los personajes Amita, Sabal y TenientePagan no deben recibir ningún disparo manteniendo su contador de vida quitada a “zero”.

Resultado

Para el problema 10 se obtienen muchos planes que no alcanzan los objetivos que se han definido. A continuación, se muestran dos planes en los que el panificador no encontró solución para el problema 10. Para generar estos planes se ejecutó el dominio con una probabilidad de aleatoriedad del 70%.

Plan 1

AMITA consulta a AJAY si sería capaz de MATARAPAGAN, uno de los líderes que tiene oprimido al pueblo de Kirat, AJAY sin dudarlo se embarca en la misión

SABAL desea recuperar la cultura RELIGIOSA de Kirat, por lo que va a intentar LIBERARLAIGLESIA de Kirat, que controla la Guardia Real

El miserable TENIENTEYUMA quiere conseguir el ANTIDOTO que podría acabar con su líder YUMA

TENIENTEYUMA se dirige de FORTALEZA_YUMA a BANAPUR para atacar a un miembro de la Senda Dorada

AMITA desea recuperar la cultura POLITICA de Kirat, por lo que va a intentar

DESTRUIRLAFABRICA de Kirat, que controla la Guardia Real
El miserable TENIENTEPAGAN quiere conseguir el ARMAMENTO que podría acabar con su líder PAGAN
El miserable TENIENTEPAUL quiere conseguir el EXPLOSIVO que podría acabar con su líder PAUL
TENIENTEPAGAN se dirige de FORTALEZA_PAGAN a la FABRICA de SHANATH dado que allí se sitúa el eficaz ARMAMENTO que necesita
TENIENTEPAUL se dirige de FORTALEZA_PAUL a BANAPUR para atacar a un miembro de la Senda Dorada
AJAY sabe que si quiere MATARAPAGAN necesitara tener en su poder un fuerte ARMAMENTO, por lo que quiere conseguirlo antes de ir a por PAGAN
En la FABRICA de SHANATH. TENIENTEPAGAN coge un formidable ARMAMENTO
TENIENTEPAUL dispara AMITA con su AK-74 en BANAPUR
... to be continued.

Plan 2

SABAL le pide a AJAY que le ayude a LIBERARLAIGLESIA de Kirat, AJAY decidido a ayudar a su compañero a recuperar la cultura RELIGIOSA de kirat acepta la misión
TENIENTEYUMA se dirige de FORTALEZA_YUMA a BANAPUR para atacar a un miembro de la Senda Dorada
TENIENTEPAUL se dirige de FORTALEZA_PAUL a BANAPUR para atacar a un miembro de la Senda Dorada
TENIENTEPAUL dispara SABAL con su AK-74 en BANAPUR
... to be continued.

A continuación, se muestra uno de los planes encontrados por el planificador para el problema 10. En concreto este problema se ha obtenido aplicando una probabilidad de aleatoriedad del 0.001%.

Plan 3

SABAL le pide a AJAY que le ayude a LIBERARLAIGLESIA de Kirat, AJAY decidido a ayudar a su compañero a recuperar la cultura RELIGIOSA de kirat acepta la misión
AMITA desea recuperar la cultura POLITICA de Kirat, por lo que va a intentar DESTRUIRLAFABRICA de Kirat, que controla la Guardia Real
El ruin PAGAN quiere el potente ARMAMENTO para salvar su vida
El ruin PAUL quiere el potente EXPLOSIVO para salvar su vida
El miserable TENIENTEYUMA quiere conseguir el ANTIDOTO que podría acabar con su líder YUMA
El miserable TENIENTEPAUL quiere conseguir el EXPLOSIVO que podría acabar con su líder PAUL
TENIENTEYUMA se dirige de FORTALEZA_YUMA a BANAPUR para atacar a un miembro de la Senda Dorada
SABAL dispara TENIENTEYUMA con su MP44 en BANAPUR
SABAL dispara TENIENTEYUMA con su MP44, TENIENTEYUMA no aguanta más disparos y muere en BANAPUR
SABAL cansado de la opresión bajo la que se encuentran los ciudadanos de Kirat, va a intentar MATARAPPAUL acabando así con uno de los líderes de la dictadura

AJAY se dirige de BANAPUR a la IGLESIA de VARSHAKOT para llevar a cabo su misión LIBERARLAIGLESIA

SABAL sabe que si quiere MATARAPPAUL necesitara tener en su poder un fuerte EXPLOSIVO, por lo que quiere conseguirlo antes de ir a por PAUL

El miserable TENIENTEPAUL quiere conseguir el ARMAMENTO que podría acabar con su líder PAGAN

En la IGLESIA de VARSHAKOT, AJAY utiliza C4 para acabar con todos los soldados AMITA se dirige de BANAPUR a la FABRICA de SHANATH para llevar a cabo su misión DESTRUIRLAFABRICA

En la FABRICA de SHANATH, AMITA utiliza su MP5 para acabar con un soldado

En la FABRICA de SHANATH, AMITA utiliza su MP5 para acabar con un soldado

SABAL se dirige de BANAPUR a la IGLESIA de VARSHAKOT dado que allí se sitúa el eficaz EXPLOSIVO que necesita

En VARSHAKOT, AJAY consigue que la IGLESIA quede LIBERADA completando así su misión LIBERARLAIGLESIA

SABAL consulta a AJAY si sería capaz de MATARAYUMA, uno de los líderes que tiene oprimido al pueblo de Kirat, AJAY sin dudarlo se embarca en la misión

AJAY sabe que si quiere MATARAYUMA necesitara tener en su poder un fuerte ANTIDOTO, por lo que quiere conseguirlo antes de ir a por YUMA

TENIENTEPAUL se dirige de FORTALEZA_PAUL a la IGLESIA de VARSHAKOT dado que allí se sitúa el eficaz EXPLOSIVO que necesita

En la IGLESIA de VARSHAKOT. TENIENTEPAUL coge un formidable EXPLOSIVO

TENIENTEPAUL dispara AJAY con su AK-74 en VARSHAKOT

SABAL dispara TENIENTEPAUL con su MP44 en VARSHAKOT

AJAY se dirige de VARSHAKOT a la TORRE de TIRTHA dado que allí se sitúa el eficaz ANTIDOTO que necesita

TENIENTEPAUL se dirige de VARSHAKOT a FORTALEZA_PAUL porque quiere darle EXPLOSIVO a su líder PAUL

En FORTALEZA_PAUL, TENIENTEPAUL le da EXPLOSIVO a su líder PAUL

PAUL consigue salvar su vida destruyendo el abrumador EXPLOSIVO, Destruído el artefacto, la Senda Dorada ya no podrá acabar con el

En la TORRE de TIRTHA, AJAY utiliza C4 para acabar con todos los soldados

TENIENTEPAUL se dirige de FORTALEZA_PAUL a TIRTHA para atacar a un miembro de la Senda Dorada

AJAY dispara TENIENTEPAUL con su AK-47, TENIENTEPAUL no aguanta más disparos y muere en TIRTHA

En TIRTHA, AJAY consigue que la TORRE quede DESTRUIDA completando así la misión de DESTRUIRLATORRE

En la TORRE de TIRTHA, AJAY coge un mortífero ANTIDOTO

AJAY se dirige de TIRTHA a FORTALEZA_YUMA con el letal ANTIDOTO para acabar con YUMA

En FORTALEZA_YUMA, AJAY mata YUMA utilizando el gran ANTIDOTO

SABAL consulta a AJAY si sería capaz de MATARAPAGAN, uno de los líderes que tiene oprimido al pueblo de Kirat, AJAY sin dudarlo se embarca en la misión

AJAY sabe que si quiere MATARAPAGAN necesitara tener en su poder un fuerte ARMAMENTO, por lo que quiere conseguirlo antes de ir a por PAGAN

En SHANATH, AMITA consigue que la FABRICA quede DESTRUIDA completando así su misión DESTRUIRLAFABRICA

AJAY se dirige de FORTALEZA_YUMA a la FABRICA de SHANATH dado que allí se sitúa el eficaz ARMAMENTO que necesita

En la FABRICA de SHANATH, AJAY coge un mortífero ARMAMENTO

AJAY se dirige de SHANATH a FORTALEZA_PAGAN con el letal ARMAMENTO para acabar con PAGAN

PAGAN dispara AJAY con su M16, AJAY no aguanta más disparos y muere en FORTALEZA_PAGAN

En FORTALEZA_PAGAN, PAGAN coge ARMAMENTO del cuerpo sin vida de AJAY PAGAN consigue salvar su vida destruyendo el abrumador ARMAMENTO. Destruído el artefacto, la Senda Dorada ya no podrá acabar con el

End.

Datos Obtenidos para el Problema 10

aleatoriedad (0,001%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	Si	Si	Si	Si	-
Longitud	45	45	45	45	45
Diversidad	-	0	0	0	0
aleatoriedad (20%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	No	No	No	Si	-
Longitud	11	32	36	51	-
Diversidad	-	27	26	59	-
aleatoriedad (50%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	No	No	No	No	-
Longitud	7	9	22	8	-
Diversidad	-	10	19	18	-
aleatoriedad (70%)	Plan-1	Plan- 2	Plan--3	Plan- 4	Media
Encuentra Plan	No	No	No	No	-
Longitud	12	4	12	26	-
Diversidad	-	16	12	30	-

Tabla 91: Datos Problema 10

6.3. Análisis de los Resultados

Tras obtener los datos descritos en el plan de pruebas para cada uno de los problemas con los que hemos probado el domino en el planificador, podemos sacar varias conclusiones acerca de los datos que se recogen. A primera vista, se puede observar en cada una de las tablas donde almacenamos estos datos que sus valores están muy condicionados por la probabilidad de aleatoriedad que se aplica en cada ejecución de los problemas.

A continuación, pasamos a comentar el comportamiento de cada uno de los datos obtenidos para los problemas definidos.

- **Encuentra Plan:** Como se puede ver los problemas más sencillos encuentran solución en todas las ejecuciones realizadas incluidas las ejecuciones realizadas con probabilidades de aleatoriedad alta. Sin embargo, a medida que se va aumentando la dificultad de los problemas, encontrar o no un plan está muy ligado a aplicar probabilidad de aleatoriedad alta o baja, encontrando un plan cuando la aleatoriedad no es más alta de 20% y costándole al planificador encontrar un plan cuando la aleatoriedad es igual o mayor a al 50%. Algunos problemas en los que las metas son difíciles de cumplir, como por ejemplo en el problema 9 y el problema 10, el planificador solo encuentra plan cuando la probabilidad de aleatoriedad es de 0,001%

Cabe destacar que, para probar el correcto funcionamiento del dominio implementado se han realizado unas ejecuciones con una probabilidad bajísima de aleatoriedad (0.001%) para poder observar cómo se comporta el dominio en el planificador cuando éste es totalmente determinista, y como se puede ver en los datos obtenidos el planificador siempre genera un plan.

- **Longitud:** La longitud de los problemas crece a medida que se aumentan las metas de los problemas. Por ejemplo, del problema 1 al problema 4 la longitud aumenta para cada problema y lo mismo pasa del problema 5 al problema 7. Para los problemas 8, 9 y 10 este comportamiento puede variar ya que en estos problemas se han realizado diversas mezclas de los problemas 1 al 4 y los problemas 5 al 7.

El comportamiento de la longitud de los planes generados también crece en las distintas ejecuciones que se realizan para un mismo problema, cuando la probabilidad de aleatoriedad con la que se ejecuta el problema se aumenta. Este comportamiento se puede apreciar en todos los problemas planteados.

Por último, es importante destacar que para los problemas en los que no hay probabilidad de aleatoriedad la longitud del problema es constante en todas las ejecuciones. También se debe aclarar antes de pasar a explicar el comportamiento del siguiente dato, que solo se ha calculado la media de la longitud de los planes generados, cuando se han encontrado mínimo 2 planes de las 4 ejecuciones realizadas para una misma probabilidad de aleatoriedad.

- **Diversidad:** El comportamiento que sigue el dato de la diversidad solo se puede apreciar cuando los problemas se ejecutan con probabilidad de aleatoriedad,

ya que si los problemas se ejecutan sin probabilidad de presentar aleatoriedad la diversidad que presenta un plan frente a otro en dos ejecuciones distintas del mismo problema, es nula.

Como ocurría con la longitud, la diversidad aumenta a medida que se ejecutan los problemas con probabilidades de aleatoriedad más altas.

Por último, igual que hemos aclarado para el dato de la longitud, la media de la diversidad solo ha sido calculada cuando se han obtenido como mínimo dos diversidades de un plan frente a otro en las que ambos planes en los que se calcula la diversidad han encontrado la solución al problema.

En cuanto a la coherencia que presentan los planes generados en los distintos apartados de resultados de los diferentes problemas se han mostrado algunos planes de los obtenidos en la experimentación. Como ya se ha dicho anteriormente, para hacer un seguimiento más fácil las historias, se han marcado en verde las acciones que hacen que las metas del problema propuesto se completen. Además, haciendo una lectura de las narrativas se puede concluir que las historias obtenidas han sido las esperadas y además se puede añadir que realizando una planificación con aleatoriedad se pueden obtener gran variedad de historias distintas para un mismo problema manteniendo la coherencia que se esperaba de todas ellas.

7. Gestión del proyecto

En este apartado se detalla la planificación inicial que se hizo del trabajo y la planificación final que realmente ha tenido. Además, en esta sección se muestra el presupuesto del trabajo.

7.1. Planificación

A mediados del mes de enero de 2017, una vez decidido que el Trabajo Fin de Carrera iba a ser el descrito en el presente documento, se realizó una planificación de cuál sería el ciclo de desarrollo que iba a seguir el trabajo, y cuánto tiempo se tardaría en realizar.

La idea fue que el trabajo iba a seguir un ciclo de desarrollo en espiral. En cuanto a la duración se previó, que el tiempo estimado en realizar el trabajo sería de 4 meses, empezando el día 1 de febrero de 2017 y se acabando el día 31 de mayo de ese mismo año. En estos 4 meses se invertiría una media de 20 horas a la semana, de lunes a viernes, es decir se dedicarían 4 horas diarias al trabajo.

El ciclo de desarrollo en espiral consiste en ir realizando el trabajo en varias iteraciones, y en cada iteración realizar unas determinadas actividades. Este modelo permite desarrollar versiones incrementales del trabajo. Durante la primera iteración, la versión del trabajo es sencilla. Sin embargo, el objetivo es que en la última iteración el trabajo sea lo más completo. En la siguiente ilustración se puede ver a grandes rasgos la idea de este ciclo de desarrollo en espiral.



Ilustración 6: Ciclo de Desarrollo en Espiral

La razón de elegir un desarrollo en espiral para este trabajo, se ha basado en la experiencia que tenía en otros trabajos similares, aunque de menor dimensión, con el lenguaje de programación PDDL.

Las actividades que se realizan en cada iteración, son en orden, las siguientes:

- **Determinar Objetivos:** En esta primera actividad, se enumeran los objetivos de la iteración.
- **Análisis de riesgo:** Una vez tenemos los objetivos, antes de ponernos a desarrollar el código, analizamos cómo podemos conseguirlo y cómo no debemos hacerlo, para evitar conflictos con lo que ya tenemos, si es que ya tenemos algo.

- **Desarrollo y pruebas:** En esta tercera actividad, escribimos el código del dominio, necesario para cumplir con los objetivos, en esta fase podemos añadir al dominio, objetos, predicados, constantes y acciones. Una vez escrito el código del dominio, pasamos a probarlo, para ello nos creamos problemas en los cuales podamos ver el funcionamiento del dominio generado, y de esta manera comprobamos que se cumplen los objetivos de la iteración actual.
- **Planificación:** En esta última actividad, se evalúa la versión del trabajo creada y si se considera necesario se comienza a planificar la siguiente iteración.

En la siguiente imagen se muestra el diagrama de Gantt que ilustra cuál fue la planificación inicial del trabajo.

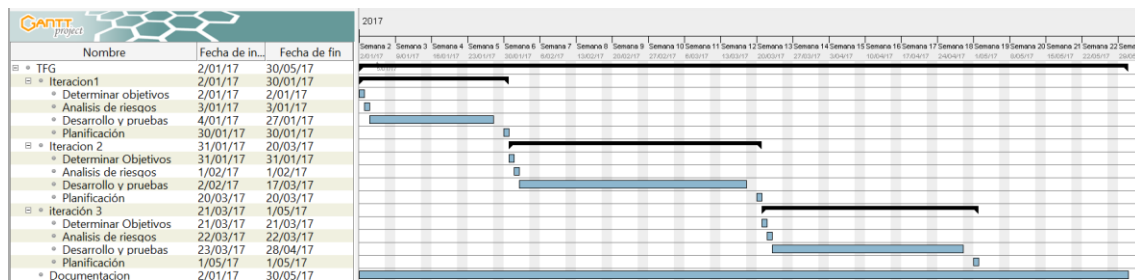


Ilustración 7: Gantt Planificación Inicial

Como se puede observar en esta planificación inicial el trabajo pasaría por 3 iteraciones del ciclo de desarrollo en espiral. Durando cada iteración 21, 35 y 20 días respectivamente y siendo la actividad de Desarrollo y Pruebas, la más pesada en cada iteración, en cuanto a tiempo.

7.2. Planificación final

En la siguiente imagen se muestra el diagrama de Gantt que ilustra cual ha sido la planificación final del trabajo.

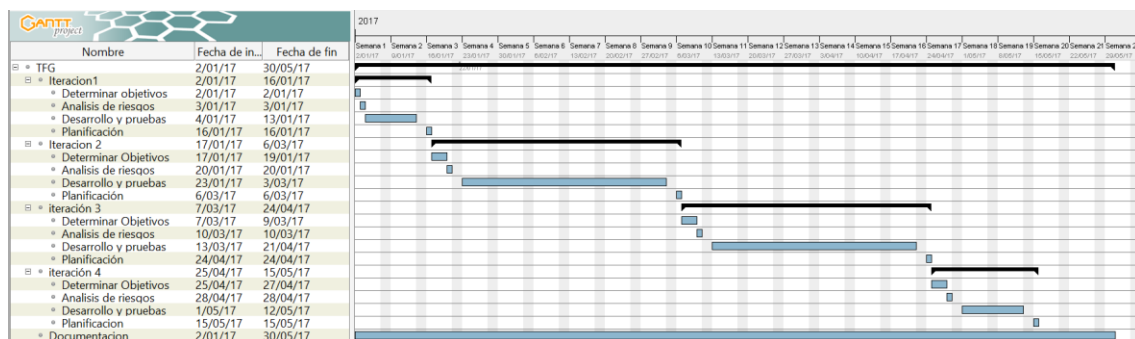


Ilustración 8: Gantt Planificación final

Como se puede ver, el trabajo se ha realizado en general como se planificó inicialmente. De hecho, la duración del trabajo no se ha visto alterada, y tal y como se predijo las actividades de Desarrollo y Pruebas han sido las más largas. Pero sí es cierto que ha habido algunas diferencias como cabía de esperar en un trabajo como éste. Una de las grandes diferencias entre la planificación inicial y la final es que el número

de iteraciones del ciclo de desarrollo en espiral por el que ha pasado el trabajo ha sido de cuatro, en vez de tres como se planeó en un primer momento. La razón de esta variación es que, en la primera iteración, los objetivos que se determinaron se quedaron un poco cortos, pudiendo alcanzarlos de forma rápida, pero dejando mucho trabajo por hacer.

Otra de las diferencias que podemos encontrar es que finalmente las actividades de Determinar Objetivos que se han llevado a cabo en cada iteración han sido un poco más duraderas.

7.3. Presupuesto

En este apartado se va a realizar el desglose de un posible presupuesto del trabajo realizado. Se tendrán en cuenta de una manera realista el coste del personal involucrado en la realización del proyecto, el coste de los materiales utilizados a lo largo de la realización del trabajo y los costes indirectos.

7.3.1. Salarios

A continuación, vamos a mostrar el coste total producido por los empleados del proyecto, teniendo en cuenta los costes de IRPF, seguridad social y salario individual en función del perfil de cada empleado.

Se considera que este proyecto puede ser realizado por dos empleados, un jefe de proyecto y un programador.

Perfil	Sueldo neto mensual(€)	IRPF (€)	Seguridad Social (€)	Sueldo mensual bruto (€)	Coste Mensual (€)	Coste por Hora (€)
Jefe de Proyecto	3.000,00	600,00	900,00	4.500,00	5.727,27	35,79
Programador	2.000,00	400,00	600,00	3.000,00	3.818,18	23,86

Tabla 92: Salarios Trabajadores

En estos costes, se ha tenido en cuenta que el IRPF es del 20% del sueldo mensual neto, así como se ha tenido también en cuenta que la seguridad social supone un 30% del sueldo mensual neto.

Por otra parte, para calcular el coste por hora de cada empleado, se ha considerado que el sueldo mensual bruto de cada empleado debe ser pagado 14 veces al año (se tienen en cuenta las pagas extras). Además, cada empleado trabajara únicamente 11 meses al año (se tiene en cuenta un mes de vacaciones). Por último, hay que especificar que con este sueldo un empleado suele trabajar 40 horas semanales.

7.3.2. Personal

En la siguiente tabla se muestran el coste del personal en el proyecto, en función de las horas empleadas en el mismo:

Actividad	Jefe de Proyecto	Programador	Total (€)
Determinar Objetivos	10	40	1312,30
Análisis De Riesgos	5	16	560,71
Desarrollo y pruebas	0	312	7.444,32
Planificación	10	16	739,66
Total	25	384	10.056,99

Tabla 93: Coste Personal

7.3.3. Materiales

Descripción	Coste por Unidad (€)	Unidades necesarias	Coste total Sin Iva (€)	Coste total Con Iva (€)	Coste imputable al proyecto (€)
Ordenador	889,00	2	1778,00	2151,38	179,28
Microsoft Office 365 Personal	41,31	2	82,62	99,98	33,32
SAYPHI	0	2	0	0	0
Total			1860,62	2251,36	212,60

Tabla 94: Coste Materiales

Para calcular los costes imputables al proyecto, de cada material se ha considerado que cada ordenador tendrá una vida como mínimo de 4 años y que la licencia de Microsoft Office 365 Personal es válida únicamente para 1 año. Además, hay que tener en cuenta, para calcular estos costes, que este trabajo se realiza en un periodo de 4 meses.

7.3.4. Gastos Indirectos

La siguiente tabla muestra los costes indirectos, que se obtienen con el 15% de los costes directos. Considerando que los costes directos son igual a la suma de los costes de personal y de materiales calculados en los apartados anteriores.

Descripción	Coste Directo (€)	Coste Indirecto (€)
Costes Indirectos	6.534,54	1.540,43

Tabla 95: Gastos Indirectos

7.3.5. Resumen

En la siguiente tabla se puede ver cómo quedaría el presupuesto considerando los costes calculados anteriormente y teniendo en cuenta que se quiere aplicar un beneficio del 10%.

Descripción	Coste (€)
Coste Empleados	10.056,96
Coste Materiales	212,60
Coste indirectos	1.540,43
Coste Total	11.810,03
Precio Total (Sin IVA, Beneficio 10%)	12.991,03

Tabla 96: Presupuesto

7.4. Entorno Socio-Económico.

En este apartado cabe destacar la existencia de dos grupos de investigación que trabajan o han trabajado en un campo bastante similar de la Inteligencia Artificial al presentado en este Trabajo. Los grupos que se presentan en este apartado han llevado a cabo proyectos relacionados con la generación de contenido de entretenimiento mediante Inteligencia Artificial.

Grupo de Investigación de la Universidad de Teesside.

Este grupo de investigación de la Universidad de Teesside está liderado por el Dr. Julie Porteous. Este grupo de investigación es famoso por su trabajo realizado sobre la Integración de la investigación en la Narrativa Interactiva (IRIS, Integrating Research in Interactive Storytelling). El trabajo de este grupo ha logrado grandes avances en la comprensión de la narrativa interactiva y el desarrollo de las diferentes tecnologías implicadas en este campo de estudio. Actualmente este grupo de investigación está trabajando en la creación de un cuentacuentos interactivo.

Liquid Narrative Group

Este grupo de la Universidad Estatal de Carolina de Norte trabajó creando un procedimiento para generar contenido. Su objetivo era poder crear contenidos de videojuegos u otros entornos virtuales de entretenimiento. Este proyecto fue acabado

en 2005 y consiguió generar un proceso que permitía crear y contar historias narrativas automáticamente.

7.5. Marco Regulador.

Tras consultar las diferentes leyes que podrían afectar al proyecto se puede concluir que no existe ninguna ley que se aplique a este Trabajo Fin de Grado.

8. Conclusiones

8.1 Comentarios finales.

El principal objetivo del proyecto era elaborar un sistema que permitiera crear gran variedad de historias narrativas mediante planificación automática. Tras analizar los resultados obtenidos en la fase de experimentación podemos afirmar que el objetivo principal ha sido alcanzado.

Desde el punto de vista narrativo las historias que crea el sistema son interesantes y coherentes y aunque únicamente el sistema generara una secuencia de los eventos que ocurren dentro de una historia, estas secuencias de eventos podrían ser usadas para elaborar:

- La narración de un libro
- El guion de una película
- La trama de un videojuego

Lo más difícil del trabajo ha sido dominar el lenguaje PDDL que, aunque estaba un poco familiarizado con él, no lo dominaba completamente y me costó mucho tiempo acostumbrarme a él. También he encontrado ciertas dificultades a la hora de modelar el dominio de planificación, cambiando varias decisiones de modelado durante el trabajo.

8.2. Líneas futuras.

Hay muchas posibilidades para mejorar este trabajo fin de grado. En este apartado comentamos algunas de ellas.

Para mejorar la calidad de las historias narrativas que se generan, se podría extender el código del dominio implementado a PDDL3.0. Permitiéndonos realizar algunas mejoras en las narraciones utilizando “fluents” en el dominio, para modelar distintos atributos de los elementos de las historias como, por ejemplo, la vida de los personajes, el poder de las armas y el número de soldados que defienden las construcciones de Kirat. También se podrían mejorar las traducciones de las acciones del dominio para generar historias más fáciles de leer y que resulten menos repetitivas.

Por último, se podrían realizar trabajos adicionales para mejorar la experiencia de los usuarios como, por ejemplo:

- Creando una interfaz para que los usuarios puedan generar sus propias historias haciendo ellos los problemas.
- Implementando un software de procesamiento de lenguaje natural que lea en voz alta las historias.

9. Referencias

- [1] Ghallab M., Howe, A., Knoblock, C., McDermott, D., Ram, A., Veloso, M., Weld, D. & Wilkins, D. (1998). PDDL – The Planning Domain Definition Language. The AIPS-98 Planning Competition Comitee.
- [2] De la Rosa, T., García, A. & Borrajo, D. (2013). A Case-Based Approach to Heuristic Planning. *Applied Intelligence journal*, Springer Verlag, Vol. 39, pp.184-201.
- [3] George, A R. (2014). Narrative Generation in Entertainment: Using Artificial Intelligence Planning. Paper, trabajo científico.
- [4] Marrella, A. & Vassos, S. Story Generation in PDDL Using Character Moods: A Case Study on Iliad's First Book. Universidad Sapienza de Roma. Paper, trabajo científico.
- [5] Pérez, P. (2014). Generación Automática de Narrativas. Universidad Carlos III de Madrid, Trabajo Fin de Grado.
- [6] Blanco Álvarez, F. (2014). GENERACIÓN AUTOMÁTICA DE NARRATIVAS: Aventura Fantástica. Universidad Carlos III de Madrid, Trabajo Fin de Grado.
- [7] Haslum, P. (2012). Narrative planning: Compilations to classical planning. *Journal of Artificial Intelligence Research*, Vol. 44, pp. 383-395.
- [8] Riedl, M., & Young, R. (2010). Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research*, Vol. 39, pp. 217–268.

10. Anexo: Código

En este apartado se muestra el código del dominio implementado.

Dominio

```
(define (domain farCry)
  (:requirements :strips :typing :equality)

  (:types
    mision_SendaDorada1 mision_SendaDorada2 mision_SendaDorada3 -
mision_SendaDorada
    mision_GuardiaReal1 mision_GuardiaReal2 - mision_GuardiaReal
    mision_SendaDorada mision_GuardiaReal - mision
    arm_ligera_senda arm_ligera_guardia - arm_ligera
    arm_ligera arm_pesada arm_explosiva - arm
    personaje_guardia1 personaje_guardia2 - personaje_guardia
    personaje_senda personaje_guardia - personaje
    personaje arm construccion - object
    place_senda place_guardia place_const - place
    corriente
    contador
    estado_final
  )

  (:constants
    zero one two three - contador
    Destruida Liberada - estado_final
    politica religiosa - corriente
  )

  (:predicates
    ;CONTADOR
    (next1 ?c1 - contador ?c2 - contador)
    ;PLACES AND TRAVEL
    (at ?obj - object ?place - place)
    (numero-personajes-lugar-S ?place - place ?contador - contador)
    ;CONSTRUCCIONES
    (is ?const - construccion ?estado - estado_final)
    (construccion-soldados ?const - construccion ?c - contador)
    (construccion-soldados-muertos ?const - construccion ?c - contador)
    ;PERSONAJE
    (personaje-corriente ?personaje - personaje ?corriente - corriente)
    (personaje-vida ?personaje - personaje ?c - contador)
    (personaje-vida-quitada ?personaje - personaje ?c - contador)
    (personaje-muerto ?personaje - personaje)
    (personaje-ocupado ?personaje - personaje)
    (tiene-que-disparar ?personaje - personaje)
```

```

(criptonita ?personaje - personaje ?arm - arm_pesada)
(asesinado-por ?personaje1 - personaje ?personaje2 - personaje)
;ARMAS
(have-an-obj ?object - object ?arm - arm)
(wish-an-obj ?personaje - personaje ?arm - arm)
;MISIONES
(mision-bloqueada ?mision - mision)
(mision-complete ?m - mision)
(mision-corriente ?m - mision ?corriente - corriente)
(have-mision ?personaje - personaje ?m - mision)
(mision-relaccionada ?m - mision ?m2 - mision)
(misionS2-descripcion ?m - mision_SendaDorada2 ?const - construccion ?estado -
estado_final)
(misionS3-descripcion ?m - mision_SendaDorada3 ?personaje - personaje)
(misionG2-descripcion ?m - mision_GuardiaReal2 ?personaje - personaje)
)

```

```

;MISIONES
;ACC-001
(:action mandar-mision1
  :parameters (
    ?personaje1 ?personaje2 - personaje_senda
    ?place - place
    ?ms1 - mision_SendaDorada1
    ?ms2 - mision_SendaDorada2
    ?corriente - corriente)
  :precondition (and
    (not (personaje-muerto ?personaje1))
    (not (personaje-muerto ?personaje2))
    (not (= ?personaje1 ?personaje2))
    (personaje-corriente ?personaje1 ?corriente)
    (personaje-corriente ?personaje2 ?corriente)
    (have-mision ?personaje1 ?ms1)
    (not(personaje-ocupado ?personaje2))
    (not(have-mision ?personaje2 ?ms1))
    (mision-relaccionada ?ms1 ?ms2)
    (mision-corriente ?ms2 ?corriente)
    (not (mision-bloqueada ?ms2))
  )
  :effect (and
    (have-mision ?personaje2 ?ms2)
    (personaje-ocupado ?personaje2)
    (mision-bloqueada ?ms2)
  )
)
)

```



```

;ACC-002
(:action mandar-mision2
  :parameters (
    ?personaje1 ?personaje2 - personaje_senda
    ?place - place
    ?ms1 - mision_SendaDorada1
    ?ms3 - mision_SendaDorada3
    ?corriente - corriente)
  :precondition (and
    (not (personaje-muerto ?personaje1))
    (not (personaje-muerto ?personaje2))
    (not (= ?personaje1 ?personaje2))
    (personaje-corriente ?personaje1 ?corriente)
    (personaje-corriente ?personaje2 ?corriente)
    (have-mision ?personaje1 ?ms1)
    (not(have-mision ?personaje2 ?ms1))
    (not(personaje-ocupado ?personaje2))
    (mision-relaccionada ?ms1 ?ms3)
    (not (mision-bloqueada ?ms3))
  )
  :effect (and
    (have-mision ?personaje2 ?ms3)
    (personaje-ocupado ?personaje2)
    (mision-bloqueada ?ms3)
  )
)
)

```

```

;ACC-003
(:action aprender-mision1
  :parameters (
    ?personaje - personaje_senda
    ?ms1 - mision_SendaDorada1
    ?ms2 - mision_SendaDorada2
    ?corriente - corriente)
  :precondition (and
    (not (personaje-muerto ?personaje))
    (not(personaje-ocupado ?personaje))
    (have-mision ?personaje ?ms1)
    (mision-relaccionada ?ms1 ?ms2)
    (mision-corriente ?ms2 ?corriente)
    (personaje-corriente ?personaje ?corriente)
    (not (mision-bloqueada ?ms2))
  )
  :effect (and
    (have-mision ?personaje ?ms2)
    (personaje-ocupado ?personaje)
    (mision-bloqueada ?ms2)
  )
)

;ACC-004
(:action aprender-mision2
  :parameters (
    ?personaje - personaje_senda
    ?ms1 - mision_SendaDorada1
    ?ms3 - mision_SendaDorada3)
  :precondition (and
    (not (personaje-muerto ?personaje))
    (not(personaje-ocupado ?personaje))
    (have-mision ?personaje ?ms1)
    (mision-relaccionada ?ms1 ?ms3)
    (not (mision-bloqueada ?ms3))
  )
  :effect (and
    (have-mision ?personaje ?ms3)
    (personaje-ocupado ?personaje)
    (mision-bloqueada ?ms3)
  )
)

```

```

;ACC-005
(:action aprender-misionSenda-principal
  :parameters (
    ?personaje1 ?personaje2 ?personaje3 - personaje_senda
    ?ms1 - mision_SendaDorada1)
  :precondition (and
    (personaje-muerto ?personaje1)
    (personaje-muerto ?personaje2)
    (not (= ?personaje1 ?personaje2))
    (or(and(personaje-corriente ?personaje1 religiosa)(personaje-corriente
?personaje2 politica))
      (and(personaje-corriente ?personaje2 religiosa)(personaje-corriente
?personaje1 politica)))
    (have-mision ?personaje1 ?ms1)
    (have-mision ?personaje2 ?ms1)
    (not(personaje-muerto ?personaje3))
    (not (and(= ?personaje1 ?personaje3)(= ?personaje2 ?personaje3))))
  )
  :effect (and
    (have-mision ?personaje3 ?ms1)
  )
)
;CORRIENTES
;ACC-006
(:action aprender-corriente1
  :parameters (
    ?personaje1 ?personaje2 - personaje_senda
    ?ms1 - mision_SendaDorada1)
  :precondition (and
    (personaje-muerto ?personaje1)
    (not(personaje-muerto ?personaje2))
    (not (= ?personaje1 ?personaje2))
    (not(personaje-corriente ?personaje2 politica))
    (personaje-corriente ?personaje1 politica)
    (have-mision ?personaje1 ?ms1)
    (have-mision ?personaje2 ?ms1)
  )
  :effect (and
    (personaje-corriente ?personaje2 politica)
  )
)

```

```

;ACC-007
(:action aprender-corriente2
  :parameters (
    ?personaje1 ?personaje2 - personaje_senda
    ?ms1 - mision_SendaDorada1)
  :precondition (and
    (personaje-muerto ?personaje1)
    (not(personaje-muerto ?personaje2))
    (not (= ?personaje1 ?personaje2))
    (not(personaje-corriente ?personaje2 religiosa))
    (personaje-corriente ?personaje1 religiosa)
    (have-mision ?personaje1 ?ms1)
    (have-mision ?personaje2 ?ms1)
  )
  :effect (and
    (personaje-corriente ?personaje2 religiosa)
  )
)
;CONSEGUIR ARMAS

```

```

;ACC-008
(:action querer-objeto-por-mision-senda

  :parameters (
    ?personaje1 - personaje_senda
    ?personaje2 - personaje_guardia1
    ?ms3 - mision_SendaDorada3
    ?arm - arm_pesada)
  :precondition (and
    (not (personaje-muerto ?personaje1))
    (not (personaje-muerto ?personaje2))
    (have-mision ?personaje1 ?ms3)
    (misionS3-descripcion ?ms3 ?personaje2)
    (criptonita ?personaje2 ?arm)
    (not(have-an-obj ?personaje1 ?arm))
    (not(wish-an-obj ?personaje1 ?arm))
  )
  :effect (and
    (wish-an-obj ?personaje1 ?arm)
  )
)

```

```

;ACC-009
(:action querer-objeto-por-mision-guardia1
  :parameters (
    ?personaje1 - personaje_guardia1
    ?personaje2 - personaje_guardia1
    ?mg2 - mision_GuardiaReal2
    ?arm - arm_pesada)
  :precondition (and
    (not (personaje-muerto ?personaje1))
    (not (personaje-muerto ?personaje2))
    (have-mision ?personaje1 ?mg2)
    (misionG2-descripcion ?mg2 ?personaje2)
    (criptonita ?personaje2 ?arm)
    (not(have-an-obj ?personaje1 ?arm))
    (not(wish-an-obj ?personaje1 ?arm))
  )
  :effect (and
    (wish-an-obj ?personaje1 ?arm)
  )
)

;ACC-010
(:action querer-objeto-por-mision-guardia2
  :parameters (
    ?personaje1 - personaje_guardia2
    ?personaje2 - personaje_guardia1
    ?mg2 - mision_GuardiaReal2
    ?arm - arm_pesada)
  :precondition (and
    (not (personaje-muerto ?personaje1))
    (not (personaje-muerto ?personaje2))
    (have-mision ?personaje1 ?mg2)
    (misionG2-descripcion ?mg2 ?personaje2)
    (criptonita ?personaje2 ?arm)
    (not(have-an-obj ?personaje1 ?arm))
    (not(wish-an-obj ?personaje1 ?arm))
  )
  :effect (and
    (wish-an-obj ?personaje1 ?arm)
  )
)

```

```

;ACC-011
(:action coger-objeto-Senda
  :parameters (
    ?personaje - personaje_senda
    ?arm - arm
    ?place - place
    ?const - construccion
    ?estado - estado_final)
  :precondition (and
    (not (personaje-muerto ?personaje))
    (at ?personaje ?place)
    (at ?const ?place)
    (have-an-obj ?const ?arm)
    (is ?const ?estado)
  )
  :effect (and
    (not (have-an-obj ?const ?arm))
    (have-an-obj ?personaje ?arm)
  )
)

;ACC-012
(:action coger-objeto-Guardia
  :parameters (
    ?personaje - personaje_guardia2
    ?arm - arm
    ?place - place
    ?const - construccion
    ?estado - estado_final)
  :precondition (and
    (not (personaje-muerto ?personaje))
    (at ?personaje ?place)
    (at ?const ?place)
    (have-an-obj ?const ?arm)
    (not(is ?const destruida))
  )
  :effect (and
    (not (have-an-obj ?const ?arm))
    (have-an-obj ?personaje ?arm)
  )
)

```

```

;ACC-013
(:action dar-objeto-Senda
  :parameters (
    ?personaje ?personaje2 - personaje_senda
    ?arm - arm_pesada
    ?place - place)
  :precondition (and
    (not (personaje-muerto ?personaje))
    (not (personaje-muerto ?personaje2))
    (not (= ?personaje ?personaje2))
    (at ?personaje ?place)
    (at ?personaje2 ?place)
    (have-an-obj ?personaje ?arm)
    (not (wish-an-obj ?personaje ?arm))
    (wish-an-obj ?personaje2 ?arm)
  )
  :effect (and
    (not (have-an-obj ?personaje ?arm))
    (have-an-obj ?personaje2 ?arm)
  )
)

;ACC-014
(:action dar-objeto-Guardia1
  :parameters (
    ?personaje - personaje_guardia2
    ?personaje2 - personaje_guardia1
    ?arm - arm_pesada
    ?place - place)
  :precondition (and
    (not (personaje-muerto ?personaje))
    (not (personaje-muerto ?personaje2))
    (at ?personaje ?place)
    (at ?personaje2 ?place)
    (have-an-obj ?personaje ?arm)
    (wish-an-obj ?personaje2 ?arm)
  )
  :effect (and
    (not (have-an-obj ?personaje ?arm))
    (have-an-obj ?personaje2 ?arm)
  )
)

```

```

;ACC-015
(:action dar-objeto-Guardia2
  :parameters (
    ?personaje - personaje_guardia2
    ?personaje2 - personaje_guardia2
    ?arm - arm_pesada
    ?place - place)
  :precondition (and
    (not (personaje-muerto ?personaje))
    (not (personaje-muerto ?personaje2))
    (at ?personaje ?place)
    (at ?personaje2 ?place)
    (have-an-obj ?personaje ?arm)
    (wish-an-obj ?personaje2 ?arm)
  )
  :effect (and
    (not (have-an-obj ?personaje ?arm))
    (have-an-obj ?personaje2 ?arm)
  )
)

;ACC-016
(:action coger-objeto-de-personaje-muerto
  :parameters (
    ?personaje ?personaje2 - personaje
    ?arm - arm_pesada
    ?place - place)
  :precondition (and
    (not (personaje-muerto ?personaje))
    (personaje-muerto ?personaje2)
    (at ?personaje ?place)
    (at ?personaje2 ?place)
    (have-an-obj ?personaje2 ?arm)
  )
  :effect (and
    (not (have-an-obj ?personaje2 ?arm))
    (have-an-obj ?personaje ?arm)
  )
)

```



```

;ACC-017
(:action destruir-criptonita
  :parameters (
    ?personaje - personaje_guardia1
    ?arm - arm_pesada
    ?mg2 - mision_GuardiaReal2)
  :precondition (and
    (not (personaje-muerto ?personaje))
    (have-an-obj ?personaje ?arm)
    (have-mision ?personaje ?mg2)
    (misionG2-descripcion ?mg2 ?personaje)
    (criptonita ?personaje ?arm)
  )
  :effect (and
    (not (have-an-obj ?personaje ?arm))
    (mision-complete ?mg2)
  )
)

```

;ATAQUES A CONSTRUCCIONES

```

;ACC-018
(:action ataqueSendaArmaCortaEdificio
  :parameters (
    ?personaje - personaje_senda
    ?const - construccion
    ?place - place_const
    ?arm - arm_ligera
    ?c1 ?c2 ?c3 - contador)
  :precondition (and
    (not (personaje-muerto ?personaje))
    (at ?const ?place)
    (at ?personaje ?place)
    (not(is ?const Destruida))
    (not(is ?const Liberada))
    (have-an-obj ?personaje ?arm)
    (construccion-soldados ?const ?c1)
    (construccion-soldados-muertos ?const ?c2)
    (not (= ?c1 ?c2))
    (next1 ?c2 ?c3)
  )
  :effect (and
    (not(construccion-soldados-muertos ?const ?c2))
    (construccion-soldados-muertos ?const ?c3)
  )
)

```

```

;ACC-019
(:action ataqueSendaExplosivosEdificio
  :parameters (
    ?personaje - personaje_senda
    ?const - construccion
    ?place - place_const
    ?arm - arm_explosiva
    ?c1 ?c2 - contador)
  :precondition (and
    (not (personaje-muerto ?personaje))
    (at ?const ?place)
    (at ?personaje ?place)
    (not(is ?const Destruida))
    (not(is ?const Liberada))
    (have-an-obj ?personaje ?arm)
    (construccion-soldados ?const ?c1)
    (construccion-soldados-muertos ?const ?c2)
    (not (= ?c1 ?c2))
  )
  :effect (and
    (not(construccion-soldados-muertos ?const ?c2))
    (construccion-soldados-muertos ?const ?c1)
  )
)
)

```

```

;ACC-020
(:action destruir-construccion1
  :parameters (
    ?personaje - personaje_senda
    ?ms2 - mision_SendaDorada2
    ?const - construccion
    ?place - place_const
    ?c - contador
    ?estado - estado_final)
  :precondition (and
    (not (personaje-muerto ?personaje))
    (at ?const ?place)
    (at ?personaje ?place)
    (misionS2-descripcion ?ms2 ?const ?estado)
    (not(is ?const ?estado))
    (not(mision-bloqueada ?ms2))
    (construccion-soldados ?const ?c)
    (construccion-soldados-muertos ?const ?c)
  )
  :effect (and
    (mision-complete ?ms2)
    (mision-bloqueada ?ms2)
    (is ?const ?estado)
  )
)
)

```

```

;ACC-021
(:action destruir-construccion2
  :parameters (
    ?personaje - personaje_senda
    ?ms2 - mision_SendaDorada2
    ?const - construccion
    ?place - place_const
    ?c - contador ?estado - estado_final)
  :precondition (and
    (not (personaje-muerto ?personaje))
    (at ?const ?place)
    (at ?personaje ?place)
    (misionS2-descripcion ?ms2 ?const ?estado)
    (not(is ?const ?estado))
    (mision-bloqueada ?ms2)
    (have-mision ?personaje ?ms2)
    (construccion-soldados ?const ?c)
    (construccion-soldados-muertos ?const ?c)
  )
  :effect (and
    (mision-complete ?ms2)
    (is ?const ?estado)
    (not(personaje-ocupado ?personaje))
    (not (have-mision ?personaje ?ms2))
  )
)

```

```

;ACC-022
(:action destruir-construccion3
  :parameters (
    ?personaje1 ?personaje2 - personaje_senda
    ?ms2 - mision_SendaDorada2
    ?const - construccion
    ?place - place_const
    ?c - contador ?estado - estado_final)
  :precondition (and
    (not (personaje-muerto ?personaje1))
    (not (personaje-muerto ?personaje2))
    (not(= ?personaje1 ?personaje2))
    (at ?const ?place)
    (at ?personaje1 ?place)
    (misionS2-descripcion ?ms2 ?const ?estado)
    (not(is ?const ?estado))
    (mision-bloqueada ?ms2)
    (have-mision ?personaje2 ?ms2)
    (construccion-soldados ?const ?c)
    (construccion-soldados-muertos ?const ?c)
  )
  :effect (and
    (mision-complete ?ms2)
    (is ?const ?estado)
    (not(personaje-ocupado ?personaje2))
    (not (have-mision ?personaje2 ?ms2))
  )
)

```

```

;MOVIMIENTOS
;ACC-023
(:action ir-a-1
  :parameters (
    ?personaje - personaje_senda
    ?ms2 - mision_SendaDorada2
    ?const - construccion
    ?place - place
    ?place2 - place
    ?estado - estado_final
    ?c1 ?c2 ?c3 ?c4 - contador)
  :precondition (and
    (not (personaje-muerto ?personaje))
    (have-mision ?personaje ?ms2)
    (misionS2-descripcion ?ms2 ?const ?estado)
    (at ?personaje ?place)
    (at ?const ?place2)
    (not (= ?place ?place2))
    (numero-personajes-lugar-S ?place ?c2)
    (numero-personajes-lugar-S ?place2 ?c3)
    (next1 ?c1 ?c2)
    (next1 ?c3 ?c4)
  )
  :effect (and
    (not(numero-personajes-lugar-S ?place ?c2))
    (numero-personajes-lugar-S ?place ?c1)
    (not(numero-personajes-lugar-S ?place2 ?c3))
    (numero-personajes-lugar-S ?place2 ?c4)
    (not (at ?personaje ?place))
    (at ?personaje ?place2)
  )
)
)

```

```

;ACC-024
(:action ir-a-2-1
  :parameters (
    ?personaje - personaje_senda
    ?const - construccion
    ?place - place
    ?place2 - place
    ?arm - arm_pesada
    ?c1 ?c2 ?c3 ?c4 - contador)
  :precondition (and
    (not (personaje-muerto ?personaje))
    (wish-an-obj ?personaje ?arm)
    (have-an-obj ?const ?arm)
    (at ?personaje ?place)
    (at ?const ?place2)
    (not (= ?place ?place2))
    (numero-personajes-lugar-S ?place ?c2)
    (numero-personajes-lugar-S ?place2 ?c3)
    (next1 ?c1 ?c2)
    (next1 ?c3 ?c4)
  )
  :effect(and
    (not(numero-personajes-lugar-S ?place ?c2))
    (numero-personajes-lugar-S ?place ?c1)
    (not(numero-personajes-lugar-S ?place2 ?c3))
    (numero-personajes-lugar-S ?place2 ?c4)
    (not (at ?personaje ?place))
    (at ?personaje ?place2)
  )
)
)

```

```

;ACC-025
(:action ir-a-2-2
  :parameters (
    ?personaje - personaje_guardia2
    ?const - construccion
    ?place - place
    ?place2 - place
    ?arm - arm_pesada)
  :precondition (and
    (not (personaje-muerto ?personaje))
    (wish-an-obj ?personaje ?arm)
    (have-an-obj ?const ?arm)
    (at ?personaje ?place)
    (at ?const ?place2)
    (not (= ?place ?place2))
    (or
      (and(tiene-que-disparar ?personaje)(numero-personajes-lugar-S ?place zero))
      (not(tiene-que-disparar ?personaje))
    )
  )
  :effect (and
    (when (and(tiene-que-disparar ?personaje)
      (and(not(tiene-que-disparar ?personaje))))
    )
    (not (at ?personaje ?place))
    (at ?personaje ?place2)
  )
)

```



```

;ACC-026
(:action ir-a-3
  :parameters (
    ?personaje ?personaje2 - personaje_senda
    ?arm - arm_pesada
    ?place ?place2 - place
    ?c1 ?c2 ?c3 ?c4 - contador)
  :precondition (and
    (not (personaje-muerto ?personaje))
    (not (personaje-muerto ?personaje2))
    (not (= ?personaje ?personaje2))
    (have-an-obj ?personaje ?arm)
    (not(wish-an-obj ?personaje ?arm))
    (wish-an-obj ?personaje2 ?arm)
    (at ?personaje ?place)
    (at ?personaje2 ?place2)
    (not (= ?place ?place2))
    (numero-personajes-lugar-S ?place ?c2)
    (numero-personajes-lugar-S ?place2 ?c3)
    (next1 ?c1 ?c2)
    (next1 ?c3 ?c4)
  )
  :effect (and
    (not(numero-personajes-lugar-S ?place ?c2))
    (numero-personajes-lugar-S ?place ?c1)
    (not(numero-personajes-lugar-S ?place2 ?c3))
    (numero-personajes-lugar-S ?place2 ?c4)
    (not (at ?personaje ?place))
    (at ?personaje ?place2)
  )
)
)

```

```

;ACC-027
(:action ir-a-4-1
  :parameters (
    ?personaje - personaje_guardia2
    ?personaje2 - personaje_guardia2
    ?arm - arm_pesada
    ?place ?place2 - place)
  :precondition (and
    (not (personaje-muerto ?personaje))
    (not (personaje-muerto ?personaje2))
    (have-an-obj ?personaje ?arm)
    (not(wish-an-obj ?personaje ?arm))
    (wish-an-obj ?personaje2 ?arm)
    (at ?personaje ?place)
    (at ?personaje2 ?place2)
    (not (= ?place ?place2))
    (or
      (and(tiene-que-disparar ?personaje)(numero-personajes-lugar-S ?place zero))
      (not(tiene-que-disparar ?personaje))
    )
  )
  :effect (and
    (when (and(tiene-que-disparar ?personaje)
      (and(not(tiene-que-disparar ?personaje)))
    )
      (not (at ?personaje ?place))
      (at ?personaje ?place2)
    )
  )
)

```

```

;ACC-028
(:action ir-a-4-2
  :parameters (
    ?personaje - personaje_guardia2
    ?personaje2 - personaje_guardia1
    ?arm - arm_pesada
    ?place ?place2 - place)
  :precondition (and
    (not (personaje-muerto ?personaje))
    (not (personaje-muerto ?personaje2))
    (have-an-obj ?personaje ?arm)
    (wish-an-obj ?personaje2 ?arm)
    (at ?personaje ?place)
    (at ?personaje2 ?place2)
    (not (= ?place ?place2))
    (or
      (and(tiene-que-disparar ?personaje)(numero-personajes-lugar-S ?place zero))
      (not(tiene-que-disparar ?personaje))
    )
  )
  :effect (and
    (when (and(tiene-que-disparar ?personaje)
      (and(not(tiene-que-disparar ?personaje))))
    )
    (not (at ?personaje ?place))
    (at ?personaje ?place2)
  )
)

```

```

;ACC-029
(:action ir-a-5-1
  :parameters (
    ?personaje - personaje_senda
    ?personaje2 - personaje
    ?arm - arm_pesada
    ?place ?place2 - place
    ?c1 ?c2 ?c3 ?c4 - contador)
  :precondition (and
    (not (personaje-muerto ?personaje))
    (wish-an-obj ?personaje ?arm)
    (have-an-obj ?personaje2 ?arm)
    (at ?personaje ?place)
    (at ?personaje2 ?place2)
    (not (= ?place ?place2))
    (numero-personajes-lugar-S ?place ?c2)
    (numero-personajes-lugar-S ?place2 ?c3)
    (next1 ?c1 ?c2)
    (next1 ?c3 ?c4)
  )
  :effect (and
    (not(numero-personajes-lugar-S ?place ?c2))
    (numero-personajes-lugar-S ?place ?c1)
    (not(numero-personajes-lugar-S ?place2 ?c3))
    (numero-personajes-lugar-S ?place2 ?c4)
    (not (at ?personaje ?place))
    (at ?personaje ?place2)
  )
)
)

```

```

;ACC-030
(:action ir-a-5-2
  :parameters (
    ?personaje - personaje_guardia2
    ?personaje2 - personaje
    ?arm - arm_pesada
    ?place ?place2 - place)
  :precondition (and
    (not (personaje-muerto ?personaje))
    (wish-an-obj ?personaje ?arm)
    (have-an-obj ?personaje2 ?arm)
    (at ?personaje ?place)
    (at ?personaje2 ?place2)
    (not (= ?place ?place2))
    (or
      (and(tiene-que-disparar ?personaje)(numero-personajes-lugar-S ?place zero))
      (not(tiene-que-disparar ?personaje))
    )
  )
  :effect (and
    (when (and(tiene-que-disparar ?personaje))
      (and(not(tiene-que-disparar ?personaje))))
    )
    (not (at ?personaje ?place))
    (at ?personaje ?place2)
  )
)

```

```

;ACC-031
(:action ir-a-6
  :parameters (
    ?personaje1 - personaje_guardia2
    ?personaje2 - personaje_senda
    ?place ?place2 - place )
  :precondition (and
    (not (personaje-muerto ?personaje1))
    (not (personaje-muerto ?personaje2))
    (at ?personaje1 ?place)
    (at ?personaje2 ?place2)
    (not (= ?place ?place2))
    (or
      (and(tiene-que-disparar ?personaje1)(numero-personajes-lugar-S ?place zero))
      (not(tiene-que-disparar ?personaje1))
    )
  )
  :effect (and
    (when(and(tiene-que-disparar ?personaje1)
      (and(not(tiene-que-disparar ?personaje1))))
    )
    (tiene-que-disparar ?personaje1)
    (not (at ?personaje1 ?place))
    (at ?personaje1 ?place2)
  )
)

```

```

;ACC-032
(:action ir-a-7
  :parameters (
    ?personaje1 - personaje_senda
    ?personaje2 - personaje_guardia1
    ?place1 - place
    ?place2 - place
    ?arm - arm_pesada
    ?ms3 - mision_SendaDorada3
    ?c1 ?c2 ?c3 ?c4 - contador)
  :precondition (and
    (not (personaje-muerto ?personaje1))
    (have-mision ?personaje1 ?ms3)
    (misionS3-descripcion ?ms3 ?personaje2)
    (not (personaje-muerto ?personaje2))
    (at ?personaje1 ?place1)
    (at ?personaje2 ?place2)
    (not (= ?place1 ?place2))
    (criptonita ?personaje2 ?arm)
    (have-an-obj ?personaje1 ?arm)
    (numero-personajes-lugar-S ?place1 ?c2)
    (numero-personajes-lugar-S ?place2 ?c3)
    (next1 ?c1 ?c2)
    (next1 ?c3 ?c4)
  )
  :effect (and
    (not(numero-personajes-lugar-S ?place1 ?c2))
    (numero-personajes-lugar-S ?place1 ?c1)
    (not(numero-personajes-lugar-S ?place2 ?c3))
    (numero-personajes-lugar-S ?place2 ?c4)
    (not(at ?personaje1 ?place1))
    (at ?personaje1 ?place2)
  )
)
)

```

```

;ATAQUES ENTRE PERSONAJES
;ACC-033
(:action ataqueFusil-1
  :parameters (
    ?personaje1 - personaje_guardia
    ?personaje2 - personaje_senda
    ?place - place
    ?arm - arm_ligera_guardia
    ?c1 ?c2 ?c3 - contador)
  :precondition (and
    (not (personaje-muerto ?personaje1))
    (not (personaje-muerto ?personaje2))
    (at ?personaje1 ?place)
    (at ?personaje2 ?place)
    (have-an-obj ?personaje1 ?arm)
    (personaje-vida ?personaje2 ?c1)
    (personaje-vida-quitada ?personaje2 ?c2)
    (next1 ?c2 ?c3)
    (not (= ?c1 ?c3))
  )
  :effect (and
    (when (and(tiene-que-disparar ?personaje1)
      (and(not(tiene-que-disparar ?personaje1))))
    )
    (not(personaje-vida-quitada ?personaje2 ?c2))
    (personaje-vida-quitada ?personaje2 ?c3)
  )
)
)

```



```

;ACC-034
(:action ataqueFusil-2
  :parameters (
    ?personaje1 - personaje_senda
    ?personaje2 - personaje_guardia2
    ?place - place
    ?arm - arm_ligera_senda
    ?c1 ?c2 ?c3 - contador)
  :precondition (and
    (not (personaje-muerto ?personaje1))
    (not (personaje-muerto ?personaje2))
    (at ?personaje1 ?place)
    (at ?personaje2 ?place)
    (have-an-obj ?personaje1 ?arm)
    (personaje-vida ?personaje2 ?c1)
    (personaje-vida-quitada ?personaje2 ?c2)
    (next1 ?c2 ?c3)
    (not (= ?c1 ?c3))
  )
  :effect (and
    (not(personaje-vida-quitada ?personaje2 ?c2))
    (personaje-vida-quitada ?personaje2 ?c3)
  )
)

```

```

;ACC-035
(:action matar1
  :parameters (
    ?personaje1 - personaje_guardia
    ?personaje2 - personaje_senda
    ?place - place
    ?arm - arm_ligera_guardia
    ?m - mision_SendaDorada
    ?c1 ?c2 ?c3 ?c4 - contador)
  :precondition (and
    (not (personaje-muerto ?personaje1))
    (not (personaje-muerto ?personaje2))
    (at ?personaje1 ?place)
    (at ?personaje2 ?place)
    (have-an-obj ?personaje1 ?arm)
    (personaje-ocupado ?personaje2)
    (have-mision ?personaje2 ?m)
    (mision-bloqueada ?m)
    (personaje-vida ?personaje2 ?c1)
    (personaje-vida-quitada ?personaje2 ?c2)
    (next1 ?c2 ?c1)
    (numero-personajes-lugar-S ?place ?c4)
    (next1 ?c3 ?c4)
  )
  :effect (and
    (when(and(tiene-que-disparar ?personaje1))
      (and(not(tiene-que-disparar ?personaje1))))
    )
    (not(numero-personajes-lugar-S ?place ?c4))
    (numero-personajes-lugar-S ?place ?c3)
    (not(mision-bloqueada ?m))
    (personaje-muerto ?personaje2)
    (asesinado-por ?personaje2 ?personaje1)
  )
)

```

```

;ACC-036
(:action matar2
  :parameters (
    ?personaje1 - personaje_guardia
    ?personaje2 - personaje_senda
    ?place - place
    ?arm - arm_ligera_guardia
    ?c1 ?c2 ?C3 ?C4 - contador)
  :precondition (and
    (not (personaje-muerto ?personaje1))
    (not (personaje-muerto ?personaje2))
    (at ?personaje1 ?place)
    (at ?personaje2 ?place)
    (have-an-obj ?personaje1 ?arm)
    (not(personaje-ocupado ?personaje2))
    (personaje-vida ?personaje2 ?c1)
    (personaje-vida-quitada ?personaje2 ?c2)
    (next1 ?c2 ?c1)
    (numero-personajes-lugar-S ?place ?c4)
    (next1 ?c3 ?c4)
  )
  :effect (and
    (when(and(tiene-que-disparar ?personaje1))
      (and(not(tiene-que-disparar ?personaje1))))
    )
    (not(numero-personajes-lugar-S ?place ?c4))
    (numero-personajes-lugar-S ?place ?c3)
    (personaje-muerto ?personaje2)
    (asesinado-por ?personaje2 ?personaje1)
  )
)

```

```

;ACC-037
(:action matar3
  :parameters (
    ?personaje1 - personaje_senda
    ?personaje2 - personaje_guardia2
    ?place - place
    ?arm - arm_ligera_senda
    ?c1 ?c2 - contador)
  :precondition (and
    (not (personaje-muerto ?personaje1))
    (not (personaje-muerto ?personaje2))
    (at ?personaje1 ?place)
    (at ?personaje2 ?place)
    (have-an-obj ?personaje1 ?arm)
    (personaje-vida ?personaje2 ?c1)
    (personaje-vida-quitada ?personaje2 ?c2)
    (next1 ?c2 ?c1)
  )
  :effect (and
    (personaje-muerto ?personaje2)
    (asesinado-por ?personaje2 ?personaje1)
  )
)

```

```

;ACC-038
(:action matar-principal
  :parameters (
    ?personaje1 - personaje_senda
    ?personaje2 - personaje_guardia1
    ?place - place_guardia
    ?arm - arm_pesada
    ?ms3 - mision_SendaDorada3)
  :precondition (and
    (not (personaje-muerto ?personaje1))
    (have-mision ?personaje1 ?ms3)
    (misionS3-descripcion ?ms3 ?personaje2)
    (not (personaje-muerto ?personaje2))
    (at ?personaje1 ?place)
    (at ?personaje2 ?place)
    (have-an-obj ?personaje1 ?arm)
    (criptonita ?personaje2 ?arm)
  )
  :effect (and
    (not(personaje-ocupado ?personaje1))
    (not (have-mision ?personaje1 ?ms3))
    (personaje-muerto ?personaje2)
    (mision-complete ?ms3)
    (not (have-mision ?personaje1 ?ms3))
    (asesinado-por ?personaje2 ?personaje1)
  )
)
)
)

```